



Leopold Franzens Universität Innsbruck

Institut für Informatik

Medizinische Universität

Innsbruck

Universitätsklinik für HNO

CTTranslator

Computertomographie Bildbetrachtungs- und Segmentierungsprogramm

Version 1.0

Bakkalaureatsarbeit

Yusuf ÖZBEK

Innsbruck, 26.11.2008

Inhaltsverzeichnis

1. Einleitung

1.1. Aufgabenbeschreibung

Im Rahmen dieser Bakkalaureatsarbeit sollen CT-Bilder eingelesen, als 3D-Bild segmentiert und das segmentierte 3D-Modell als vtkData[2] gespeichert werden. Dabei sollen über eingelesene 2D- oder segmentierte 3D-Modelle verschiedene Bearbeitungen und Betrachtungen durchgeführt werden. Als Grundlage dienen reale CT-Bilder, in denen die Knochen- oder Weichgewebestrukturen zunächst identifiziert und über die Hounsfield-Skala[10] als 2D dargestellt werden. In einem anderen Schritt sollen diese Schnittbilder durch den Algorithmus „Marching Cubes"[11] als 3D in geeigneter Weise dargestellt werden und zuletzt sollen die dargestellten 3D-Modelle gespeichert werden. Vor der Segmentierung bzw. Speicherung der Bilder können verschiedene Bildbearbeitungsoperationen und Bildbetrachtungsaufgaben vorgenommen werden.

Hierfür wird ein Programm entwickelt, das die CT-Bilder einlesen kann und dem Benutzer die erwähnten Aufgaben ermöglicht. Da es sich bei diesem Programm um ein Segmentierungs- und Betrachtungsprogramm von CT-Bildern handelt, wird das Programm CTTranslator genannt.

2. Grundlagen

2.1. Computertomographie

Computertomographie (CT)[10] ist eine rechnerbasierte Analyse von verschiedenen Richtungen aufgenommener Röntgenaufnahmen eines Objektes, um ein dreidimensionales Bild zu erzeugen. CT ist ein schnittbildgebendes Verfahren. Mit der CT können die zu untersuchenden Körper in Schichten (Körperabschnitte) aufgenommen, am Computer zu 3D-Volumendatensätzen zusammengesetzt und schließlich betrachtet werden. So kann man z.B. Knochen, Luft (in der Lunge), Wasseransammlungen im Körper und Weichgewebe unterscheiden, da sie in CT-Bildern unterschiedliche Grautöne haben.

2.2. Funktionsprinzip

Bei der Computertomographie werden Röntgenstrahlen durch die zu untersuchenden Objekte (Körper) geschickt und auf der anderen Seite von mehreren Detektoren wieder aufgefangen. Aus dem Unterschied von ausgestrahlter zu aufgefangener Strahlung können Rückschlüsse auf das untersuchte Gewebe gezogen werden. Daraus errechnet ein Computer ein Schichtbild. Das nun vorliegende 2D-Schichtbild ist abhängig vom Gerätetyp eine Auflösung (Messmatrix) von 128^2, 256^2, 512^2 oder 1024^2 Pixel. Je nachdem, welches Ziel man mit der Untersuchung verfolgt, kann der Detailgrad der Aufnahmen durch Verringerung der Abstände der einzelnen Schichtbilder erhöht werden. Die Schichtbilder werden meist im Abstand von 0.5 mm bis 10 mm aufgenommen. Um einen Volumendatensatz zu erhalten ist es nötig, viele einzelne Schichten aufzunehmen und zu einem Volumen zusammenzusetzen. Um das Verfahren der Aufnahme zu beschleunigen, werden Mehrschicht-Spiral-Computertomographien eingesetzt, die zum einen die Röntgenröhre um den Patienten drehen und gleichzeitig den Patienten schrittweise durch die Röhre hindurch schieben, sog. Spiral-CT[13].

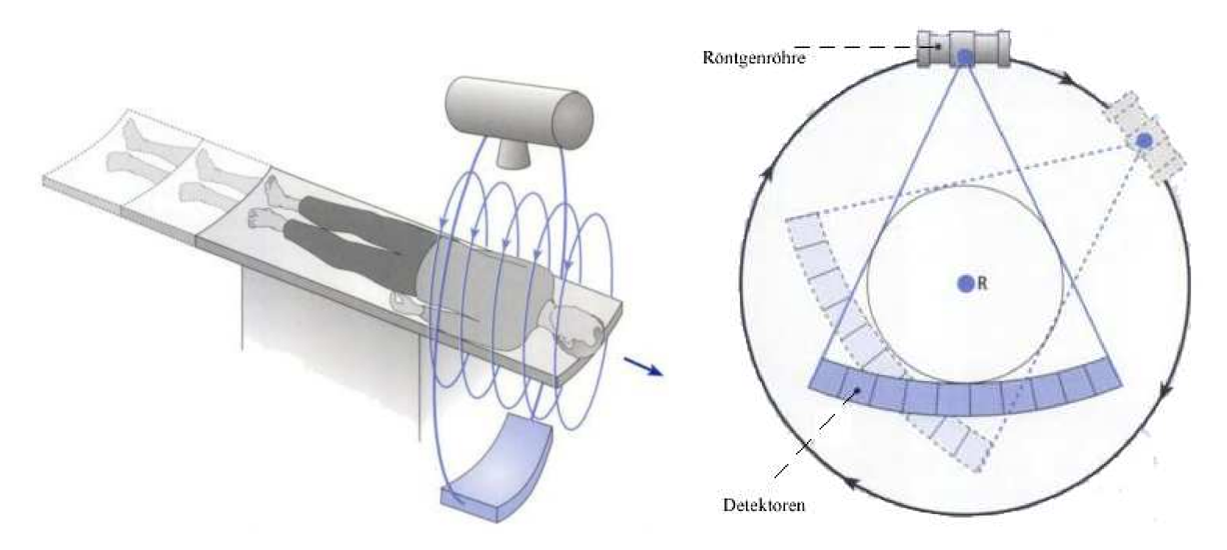


Abbildung 1: Computertomographie[14]

Mit Hilfe dieses Verfahrens ist es möglich 300 und mehr Schichtbilder in nur wenigen Sekunden aufzunehmen, während der Patient regungslos und mit angehaltenem Atem durch die Röhre geschoben wird.

2.3. Hounsfield Skala

Die Absorptionswerte des aufgenommenen Gewebes werden in Hounsfield Units (HU)[10] angegeben. Die Einheit wurde nach dem Nobelpreisträger Godfrey Hounsfield benannt, der als Vater der Computertomographie gilt. Es wurde eine Skala angelegt, bei welcher den unterschiedlichen Grauwertstufen unterschiedliche Gewebestrukturen zugeordnet werden können. Als Referenz dient Wasser mit einem Wert von 0 HU, welche den Ausgangswert für die Skala bildet. Durch Angabe eines bestimmten Grauwertes ist es allerdings noch nicht möglich, eine bestimmte Struktur zu extrahieren, da die unterschiedlichen Gewebestrukturen teils sehr ähnliche oder gar gleiche Grauwerte aufweisen.

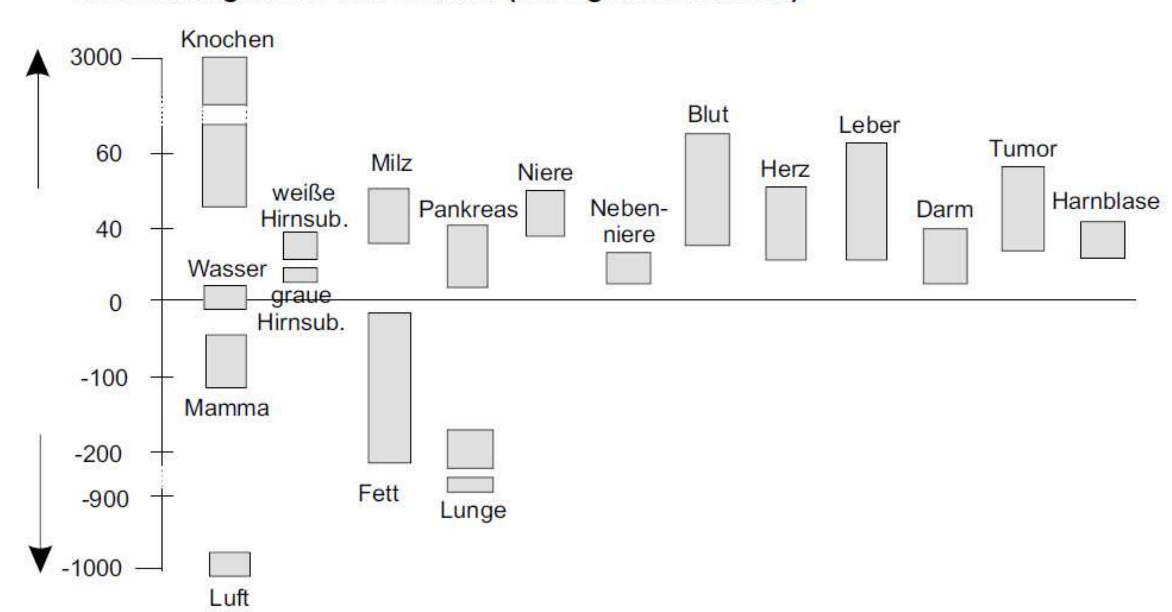


Abbildung 2: Hounsfield Skala[10]

2.4. DICOM Format

Die Grauwertbilder, die ein Computertomograph liefert, werden in der Medizin in einem speziellen Format gespeichert, das mittlerweile zum Standard für medizinische Aufnahmen geworden ist. Dieses Format nennt sich DICOM (Digital Imaging and Communications in Medicine)[12]. Das DICOM-Format beschreibt, wie die digitalen Bilder und Informationen gespeichert werden und wie medizinische Aufnahmegeräte miteinander kommunizieren können. Das Besondere ist hierbei, dass die Bilder einen Header vor dem eigentlichen Bild enthalten, in dem zusätzlich Informationen wie Patientenname, Aufnahmegerät, Bildeigenschaften, Bildgröße, Bildabstände usw. dokumentiert werden können und nicht wie bei anderen Formaten, bei denen die Zusatzinformationen in separaten Dateien gespeichert sind.

3. Material und Methoden

3.1. FLTK

Das Fast Light ToolKit (FLTK, fulltick)[5] ist Open Source Software und enthält einen eigenen GUI-Designer fluid (FLTK User Interface Designer). FLTK ist ein GUI Toolkit für 3D-Grafikprogrammierung. Es ist eine sogenannte Cross-Plattform C++ Applikation. Diese erlaubt es, ein Programm mit nur einem Quellcode für verschiedene Betriebsplattformen wie Linux, Unix, Mac OS X und Windows zu entwickeln. Die FLTK Bibliotheken liefern plattformunabhängige Schnittstellen für alle Arten von C++ Programmen. FLTK bietet auch eine große Zahl vorgefertigter grafischer Kontrollelemente (Widgets) wie Inputs- und Outputsfelder, Buttons, Textfelder usw., die nach den eigenen Bedürfnissen angepasst werden können. Ein Layout Manager erlaubt eine leichte Entwicklung von grafischen Oberflächen.

Ein FLTK Beispiel könnte folgendermaßen aussehen:

```
#include <FL/Fl.H>
#include <FL/Fl_Window.H>
#include <FL/Fl_Box.H>

int main(int argc, char **argv) {

Fl_Window *window = new Fl_Window(300,180);
Fl_Box *box = new Fl_Box(20,40,260,100,"Hello, World!");
box->box(FL_UP_BOX);
box->labelsize(36);
box->labelfont(FL_BOLD+FL_ITALIC);
box->labeltype(FL_SHADOW_LABEL);
window->end();
window->show(argc, argv);
return Fl::run();
}
```

Abbildung 3: FLTK Beispiel

3.2. VTK

Das Visualization ToolKit (VTK)[2] ist ein Open Source Softwaresystem für 3D-Computergraphik, Bildverarbeitung und Visualisierung. VTK enthält eine C++ Klassenbibliothek und verschiedene Schnittstellen, z.B. Tcl/Tk[15], Java[16] und Python[17]. Das Graphikmodel in VTK hat einen höheren Abstraktionsgrad als OpenGL oder PEX. Das heißt, es ist einfacher, Graphik- und Visualisierungsapplikationen zu erstellen. In VTK können Applikationen in C++, Tcl, Java oder Python geschrieben werden. Mit Tcl oder Python und Tk, aber auch mit Java und seinen GUI Klassenbibliotheken, sind nützliche Programme schnell erstellt. VTK ist ein echtes Visualisierungssystem und unterstützt eine große Zahl von Visualisierungsalgorithmen, wie z.B. scalar, vector, texture und volumetric methods. Ferner sind viele Bildbearbeitungsalgorithmen direkt integriert, so dass 2D-Bilder und 3D-Graphikalgorithmen gemischt verwendet werden können. Das Toolkit VTK arbeitet nach dem Pipelineprinzip. Es übergibt also das Ergebnis des einen Arbeitsschrittes an den nächsten weiter.

Ein VTK Beispiel könnte folgendermaßen aussehen:

```
#include "CTTranslator.h"
#include <vtkImageMapToColors.h>
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkDICOMImageReader.h>
#include <vtkImageActor.h>
#include <vtkLookupTable.h>

CTTranslator::CTTranslator() {

reader2D = vtkDICOMImageReader::New();
reader2D->SetDirectoryName("images/");
reader2D->UpdateWholeExtent();

this->DisplayImageAxial->show();
this->DisplayImageAxial->redraw();

renAxial = vtkRenderer::New();
renWindowAxial = vtkRenderWindow::New();
renWindowAxial->AddRenderer(renAxial);
```

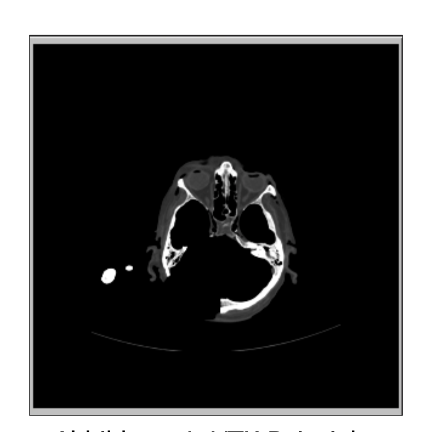


Abbildung 4: VTK Beispiel

```
DisplayImageAxial->SetRenderWindow(renWindowAxial);
DisplayImageAxial->Initialize();

hueLut = vtkLookupTable::New();
hueLut->SetTableRange (-500,1000);
hueLut->SetSaturationRange (0, 0);
hueLut->SetHueRange (0, 0);
hueLut->SetValueRange (0, 1);
hueLut->Build();

axialColors = vtkImageMapToColors::New();
axialColors->SetInput(reader2D->GetOutput());
axialColors->SetLookupTable(hueLut);

axial = vtkImageActor::New();
axial->SetInput(axialColors->GetOutput());
axial->SetOpacity(1.0);
axial->SetDisplayExtent(0,511, 0,511, 30,30);

renAxial->AddActor(axial);
}
```

3.3. CMake

CMake[9] ist eine plattformunabhängige in C++ geschriebene Open Source Software, die von der Firma Kitware[18] entwickelt wurde. Es erstellt aus einer CMakeLists.txt-Datei plattformabhängige, übersetzbare Dateien zur Ausführung von Programmen. Das Ziel dieser Software ist unter anderem den Erstellungsprozess von Software zu managen, um Compiler und Systeme benutzen zu können, die dem Programmierer bekannt sind.

Eine einfache CMakeLists.txt-Datei könnte folgendermaßen aussehen:

```
PROJECT (Beispielprojekt)

#FIND VTK
FIND_PACKAGE(VTK)
IF(USE_VTK_FILE)
INCLUDE(${USE_VTK_FILE})
ELSE (USE_VTK_FILE)
MESSAGE( FATAL_ERROR "VTK not found. Please set VTK_DIR.")
ENDIF (USE_VTK_FILE)

#FIND FLTK
FIND_PACKAGE(FLTK)
IF (USE_VTK_FILE)
INCLUDE (${FLTK_INCLUDE_DIR})
LINK_LIBRARIES( ${FLTK_LIBRARIES} ${OPENGL_LIBRARIES} )
ELSE (FLTK_FOUND)
MESSAGE( FATAL_ERROR "FLTK not found. Please set FLTK_DIR.")
ENDIF (FLTK_FOUND)

ADD_EXECUTABLE (Beispielprojekt Beispielprojekt1.cxx
Beispielprojektmain.cxx)
TARGET_LINK_LIBRARIES(Beispielprojekt vtkCommon vtkFiltering vtkGraphics
vtkHybrid vtkImaging vtkIO vtkjpeg vtkRendering vtkzlib)
```

3.4. Betriebssystem

CTTranslator ist so realisiert, dass es derzeit auf dem Betriebssystem Linux angewendet werden kann. Der mit FLTK und VTK entwickelte CTTranslator ist unter Linux (Fedora 7[19], GCC Version 4.1.2 System-Compiler[22]) mit C++ Programmierungssprache geschrieben und kann später durch Integrierung von bestimmten VTK und FLTK Bibliotheken auch unter Windows, Mac OS X, Unix usw. implementiert und benutzt werden.

Bei der Implementierung des Programms werden folgende VTK Klassen verwendet:

3.5. vtkDICOMImageReader

Die DICOM Schichtbilder werden durch vtkDICOMImageReader eingelesen. DICOM ist ein medizinisches Bild-Datei-Format, weit verbreitet zum Austausch von Daten, die aus verschiedenen Modalitäten bestehen. Der vtkDICOMImageReader könnte auch die ACR-NEMA (Vorgänger des DICOM Format für medizinische Bilder) Dateien lesen. Datensätze, im gekapseltem Format und Multi-Frames DICOM Datensätze werden nicht behandelt.

3.6. vtkRenderer/vtkRenderWindow

Ein vtkRenderer setzt eine Szene aus Lichtquellen, Kameras, Hintergrund und den erzeugten Grafikpipelines zusammen. Dabei referenziert der vtkRenderer ein vtkRenderWindow, welches im Programm einen Bereich bereitstellt, in dem der Renderer die Szene darstellt. Der Renderer wird auch benutzt, um ein Objekt, welches aus mehreren Teilen besteht, zu einem Szenegraphen zusammenzufügen. vtkRenderer führt auch Koordinatentransformation zwischen Weltkoordinaten, Ansichtskoordinaten (Computer Grafik Rendering Koordinatensystem) durch und Displaykoordinaten (die tatsächliche Bildschirmkoordinaten auf dem Bildschirm). VtkRendererWindow ist ein abstraktes Objekt um das Verhalten eines Renderings-Fensters zu spezifizieren. Es erstellt ein Fenster, in dem vtkRenderer eine Szene setzt und die geladenen Bilder darstellt.

3.7. vtkLookupTable

Die Farben werden durch eine Farbtabelle (vtkLookupTable) definiert und vom Mapper verwaltet. vtkLookupTable bildet RGBA (Rot, Grün, Blau, Alpha Transparent), Farbspezifikationen und Skalarwerte ab. Die Farbtabelle kann auch für die Generierung der Farbwerte oder Erzeugung von Farben, Sättigung, Alpha-Bereich oder einer Tabelle genutzt werden.

- SetHueRange (double min, double max) min, max ∊ [0, 1] definiert den Farbbereich der Farbtöne, 0 entspricht 0°, 1 entspricht 360°.

- SetSaturationRange(double min, double max) min, max ∊ [0, 1] definiert den Bereich der Sättigung des Farbtons, 0 entspricht der Kreismitte, 1 entspricht dem Kreisrand.

- SetValueRange(double min, double max) min, max ∊ [0, 1] definiert den Bereich der Helligkeit der Farbe, 0 entspricht Schwarz, 1 entspricht einem sehr hellen Farbton.

- SetTableRange(double min, double max) min, max ∊ [0, 1] definiert den Bereich der Skalarwerte. 0 entspricht einem minimalen Skalarwert, 1 entspricht einem maximalen Skalarwert.

3.8. vtkImageMapToColors

Der vtkImageMapToColors Filter nimmt das Eingangsbild, das vom Reader gelesen wird und bildet den ersten Bestandteil des Bildes durch eine vtkLookupTable ab. Das Resultat ist ein VTK_UNSIGNED_CHAR-artiges Bild. Wenn die vtkLookupTable nicht definiert ist oder auf NULL ersetzt wird, dann werden die Eingangsdaten übermittelt, wenn sie bereit von der VTK_UNSIGNED_CHAR-Art sind.

3.9. vtkActor/vtkProprety

Actors (vtkActor, vtkImageActor) erben von der Klasse vtkProp3D, welche für die 3D-Eigenschaften eines Objektes zuständig sind. Alle Bewegungen des Objektes werden in der vtkActor Klasse behandelt und ein Object in einer Render Szene repräsentiert. Ein vtkActor bietet Eigenschaften für Oberflächen eines Objektes (z.B. Ambient, Diffus oder Specular) durch eine assoziierte Klasse vtkProperty an. vtkProperty ist ein Objekt, das für Beleuchtung und andere Oberflächeneigenschaften (Opacitiy, Specular, Color usw.) eines geometrischen Objekts zuständig ist.

3.10. vtkCamera

vtkCamera ist eine virtuelle Kamera, die für Rendering zuständig ist. Es bietet Methoden, um die Positionen, Orientierungen, Punkt und Kontaktstelle, Blickrichtung der Ansicht zu definieren.

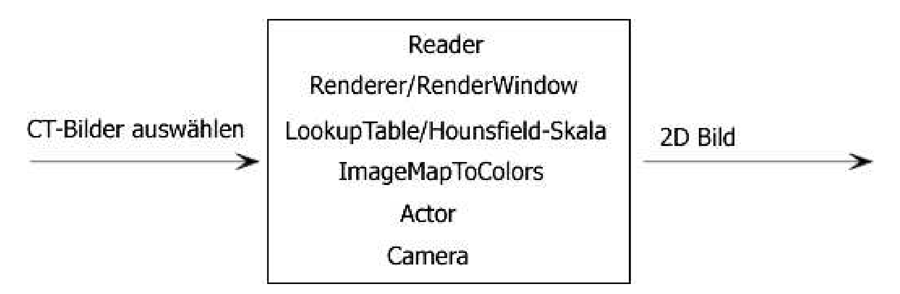


Abbildung 5: Schnittbilder einlesen

3.11. vtkMarchingCubes

Marching Cubes wurde 1987 von William E. Lorensen und Harvey E. Cline entwickelt und ist ein Algorithmus zur Berechnung von Isosurfaces in der 3D-Computergrafik. Die dazugehörige vtkMarchingCubes-Klasse enthält diesen Algorithmus. Aus einem 3D-Raster ergeben sich Würfel, welche individuell betrachtet werden, um eine Oberfläche zu ermitteln, die den Würfel teilen soll. Ein Würfel besitzt durch Querbildung acht Eckpunkte, die sich entweder innerhalb oder außerhalb der Kontur befinden und jeweils einen Index (v1, v2, v3, v4, v5, v6, v7, v8) besitzen. Somit ergeben sich $2^8 = 256$ möglichen Zuständen für den Würfel. Durch die Symmetrie des Würfels lassen sich die Zustände mit Hilfe der Rotation, Spiegelung und Inversion auf 15 reduzieren. In einer Triangle Lookup Table sind die 15 unterschiedlichen Polygone zur Teilung des Würfels

gespeichert. Die Polygone werden nach dem gleichen Muster der Eckpunkte, welche nach Innen- und Außenbereich aufgeteilt wurden, dem Würfel des Rasters zugewiesen. Anschließend wird sukzessiv zum nächsten "marschiert". Der Algorithmus terminiert, wenn der Anfangspunkt wieder erreicht wurde. Im Gesamtbild ergibt sich dann eine Oberfläche.

vtkMarchingCubes ist ein Filter, der als Eingabe ein Volumen nimmt und eine oder mehrere Isoflächen erzeugt. Um eine Isofläche erhalten zu können, müssen als Eingabe ein oder mehrere Konturwerte angegeben werden. Als zusätzliche Alternative können ein min/max skalarer Bildbereich (z.B. der Bildbereich von irgendeinem Grauwert) und die Anzahl der Konturen eingegeben werden.

Es gibt also folgende 15 Grundkonfigurationen:

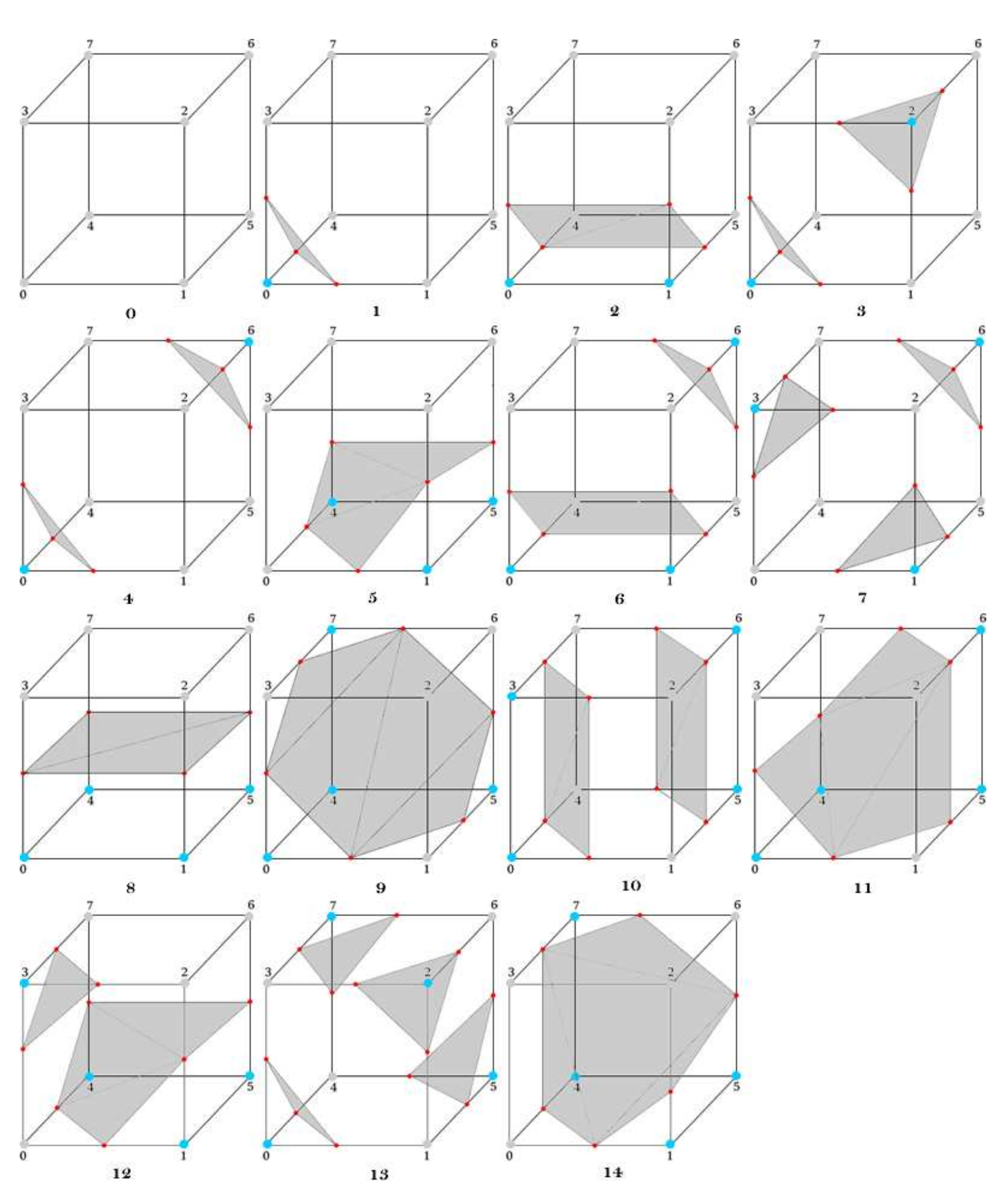


Abbildung 6: Marching Cubes Grundfunktionen

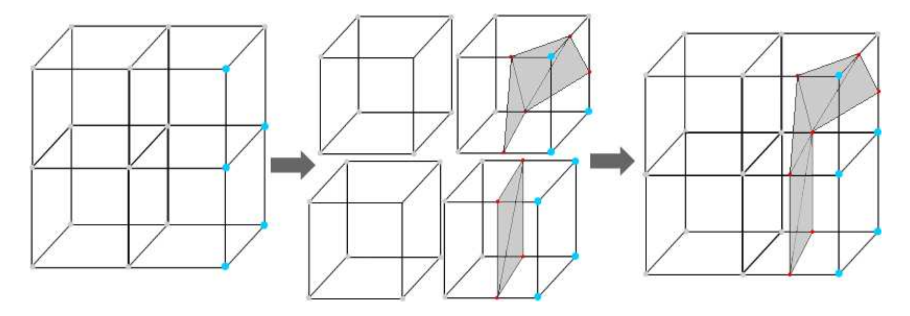


Abbildung 7: Marching Cubes Beispiel

3.12. vtkMapper

Die Abbildung zwischen den Visualisierungsdaten und den grafischen Daten wird von Mappern (vtkMapper, vtkPolyDataMapper) übernommen. Er liest Daten aus dem Datenobjekt und macht sie für das Grafikmodell verfügbar, erstellt also geometrische Primitive. Ein Mapper ist gleichzeitig die geometrische Repräsentation für einen vtkActor.

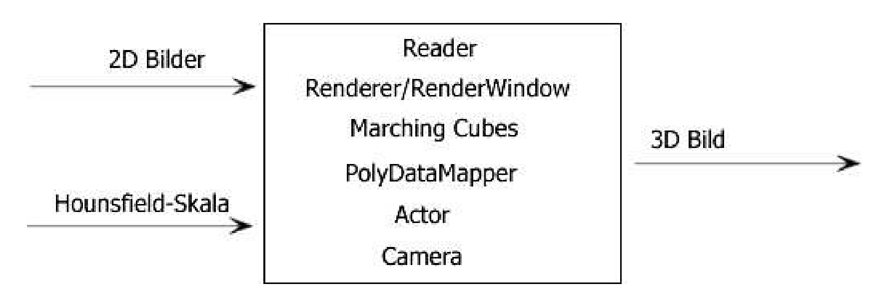


Abbildung 8: Schnittbilder segmentieren

3.13. vtkPolyDataWriter

vtkPolyDataWriter ist ein Objekt, das z.B. ein 3D-Objekt als Eingabe nimmt und sie als ASCII oder binär in VTK Format schreibt. Aber die Binärdateien, die auf ein System geschrieben werden, können möglicherweise nicht auf anderen Systemen lesbar sein.



Abbildung 9: 3D-Modell speichern

3.14. vtkWindowToImageFilter

vtkWindowToImageFilter bietet notwendige Methoden an, um die Daten (Bilder, Objekte usw.) von einem vtkRenderWindow oder vtkWindow zu lesen und sie zur Imaging-Pipeline als Eingabe zu übergeben. Damit können die Fensterinhalte als RGB oder RGBA-Pixel gelesen und in eine Datei als Bild (z.B. jpg, png usw.) gespeichert werden.



Abbildung 10: Bilder als Bildschirmfoto speichern

3.15. vtkFlRenderWindowInteractor

vtkFlRenderWindowInteractor ist eine Klasse, mit der VTK- und FLTK- Anwendungen miteinander integriert und interagiert werden können. Sie stellt einen plattformunabhängigen Interaktionsmechanismus für Maus, Tastatur, Zeitereignisse usw. zur Verfügung. vtkFlRenderWindowInteractor reagiert auf die Bewegungen des Input Devices, Buttons des Input Devices und Tastatureingaben.

3.16. vtkPointPicker

Die Klasse vtkPointPicker erlaubt durch Positionierung der Maus und durch Drücken der linken Maustaste das Picken eines Punktes. Dabei wird von der Position der Maus ein Strahl weggeschickt, der mit der Sphere geschnitten wird. Durch Picking erhält man x, y, z-Koordinaten (Identifikation eines Punktes), welche dem gepickten Punkt entsprechen.

4. Ablaufdiagramm

In diesem Abschnitt wird der Ablauf des Programms mit textueller Beschreibung des Ablaufdiagramms vorgestellt. Ablaufdiagramme werden verwendet, um die Funktionsweise eines Computerprogramms oder eines Algorithmus zu veranschaulichen. Ablaufdiagramme bilden Prozesse aus unterschiedlichen Bereichen ab und ermöglichen so eine schnelle Übersicht auch über komplexere Sachverhalte. Auf dem Pipelineprinzip aufbauend wird ein softwaretechnischer Entwurf eines Ablaufdiagramms erstellt, um zu zeigen, wie das zu entwickelnde Programm funktionieren soll und wie die Daten darin fließen.

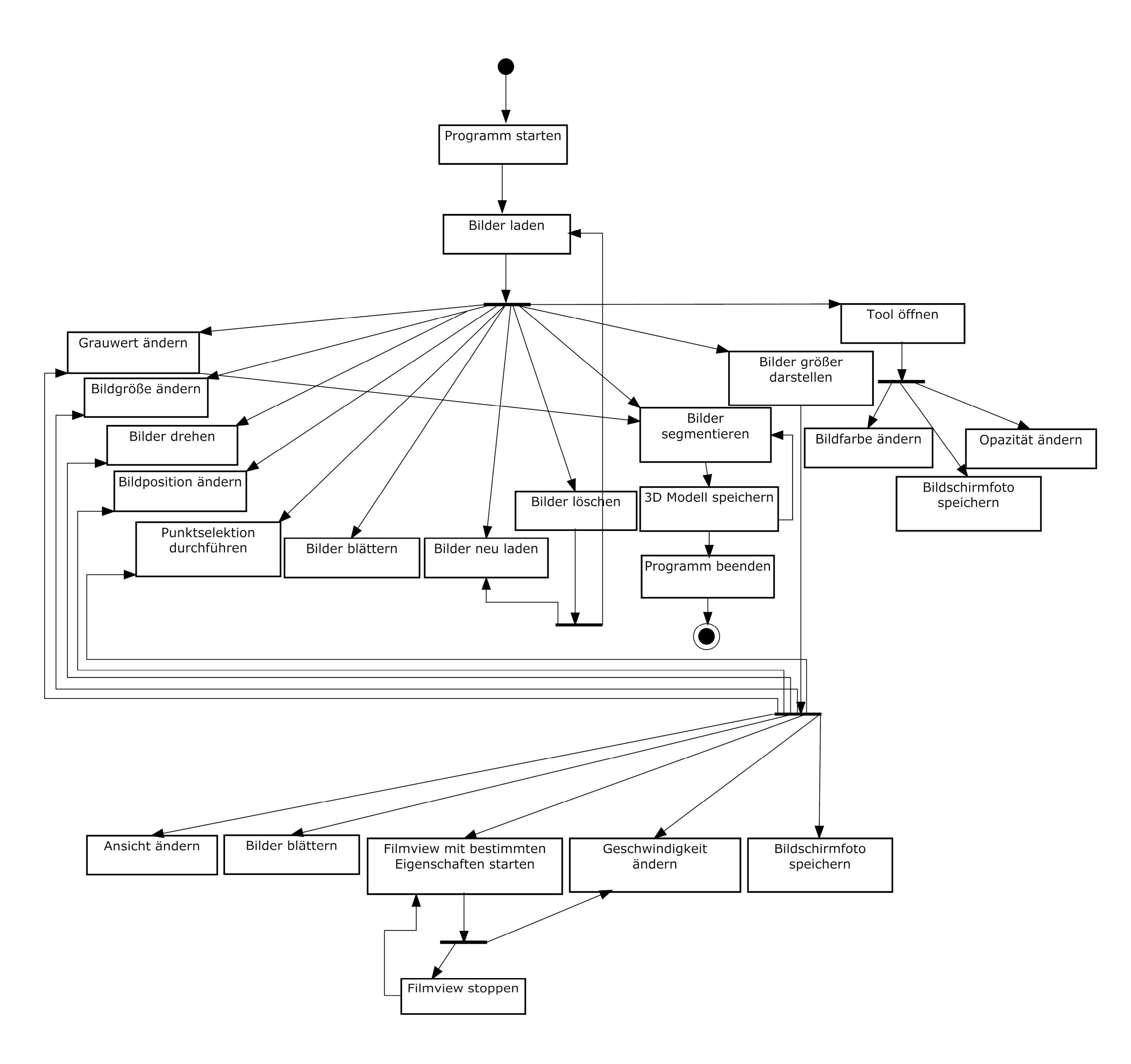


Abbildung 11: Ablaufdiagramm

Der Ablauf des Programms kann grob unter neun Einheiten unterteilt werden.

Nach erfolgreichem Start des Programms werden als erstes die gewünschten CT-Bilder eingelesen. Nachdem der Benutzer die Bilder erfolgreich geladen hat, können mit Hilfe der Mausoptionen die Aktionen: Änderung des Grauwertes, Änderung der Bildgröße, Umdrehung der Bilder, Änderung der Bildposition, Durchführung der Punktselektion zustande gebracht werden.

Der Benutzer kann dabei Bilder für jede Ansicht vorwärts oder rückwärts blättern, gleiche Bilder neu laden, Bilder löschen, Bilder segmentieren, Bilder größer darstellen oder der Toolbereich öffnen/schließen. Nachdem der Benutzer die geladenen Bilder vom Programm gelöscht bzw. entfernt hat, können diese Bilder neu geladen oder andere CT-Bilder eingelesen werden.

Die eingelesenen 2D-Bilder können auch als 3D-Modell segmentiert werden. Vor der Segmentation wird mit der Änderung des Grauwertes bei den als 2D dargestellten Bilder die gewünschten Ansichten durch den Benutzer eingestellt (z.B. Knochenstruktur oder Weichgewebe) und dann segmentiert. Nach der Segmentierung kann der Benutzer ein segmentiertes 3D-Modell als vtkData speichern, wieder zur Segmentierungsphase zurückkehren und das 3D-Modell wieder als vtkData speichern.

Nachdem der Benutzer den Toolbereich geöffnet hat, können die Aktionen: Änderung der Bildfarbe, Änderung der Bildopazität oder Speicherung der gewünschten Bilder als Bildschirmfoto ermöglicht werden.

Die geladenen Bilder können in einem größeren Fenster einzeln dargestellt werden. Die größere Darstellung ermöglicht dem Benutzer die Aktionen: Änderung der Ansicht, Blättern der Bilder, Durchführung des Filmview (Animation), Änderung der Filmview-Geschwindigkeit oder Speicherung der gewünschten Bilder als Bildschirmfoto. Beim automatischen Blättern der Bilder kann der Benutzer die Animationsgeschwindigkeit ändern oder sie stoppen.

5. Use Case Diagramm (Anwendungsfälle)

In diesem Abschnitt werden die Use Cases mit ausführlicher textueller Beschreibung und Use Case Diagramme vorgestellt. Use Cases definieren eine Interaktion zwischen Akteuren und dem betrachteten System, die stattfindet, um ein bestimmtes fachliches Ziel zu erreichen[21].

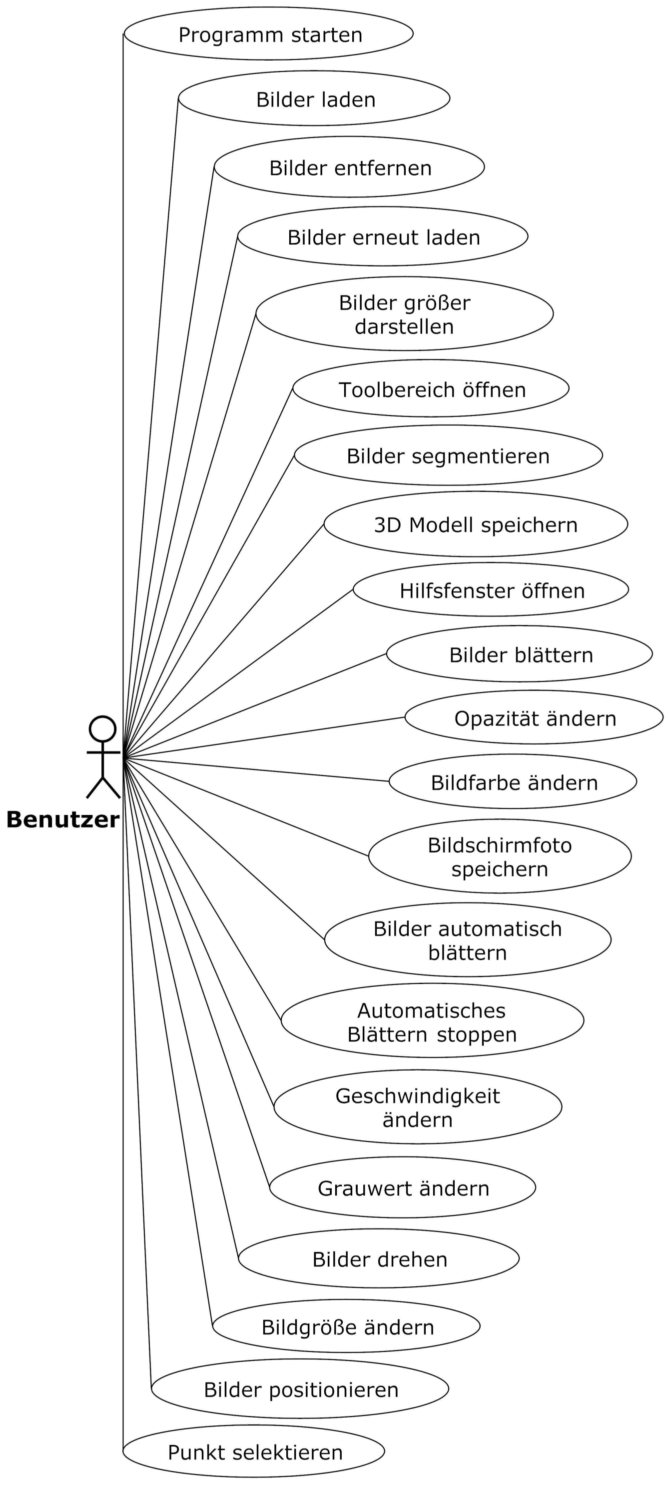


Abbildung 12: Use Case Diagramm

5.1. Programm starten

Kurzbeschreibung	Der Benutzer startet das Programm.
Akteure	Benutzer
Auslöser	Benutzer
Vorbedingung	Das Programm ist bereit zum Starten.
Nachbedingung	Es öffnet sich die Benutzeroberfläche (Hauptfenster).
Basisablauf	Der Benutzer wechselt mit „cd /CTTranslator“ in das Verzeichnis „CTTranslator“ und gibt in der Konsole oder im Terminal den Befehl „./CTTranslator“ ein.
Anmerkung	Vor dem Start des Programms müssen die Grundvoraussetzungen erfüllt werden. (siehe Benutzerhandbuch und Installations-CD unter LIESMICH.txt Datei).

5.2. Bilder laden

Kurzbeschreibung	Der Benutzer liest die Schichtbilder ein.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Das Programm ist bereits gestartet.
Nachbedingung	Bilder werden auf den Anzeigefenstern in unterschiedlichen Ansichten als 2D angezeigt und geeignete Funktionen, Buttons, Slider usw. werden aktiviert.
Basisablauf	Der Benutzer klickt auf den „Load“ Button und wählt von dem geöffneten File Chooser einen Ordner aus, in dem sich fortlaufend nummerierte CT-Bilder befinden.
Alternative Abläufe	Drücken auf die „L“ Taste.
Anmerkung	Das Programm erlaubt dem Benutzer nur CT-Bilder (fortlaufend nummeriert und mit .dcm Endung) einzulesen.

5.3. Bilder entfernen

Kurzbeschreibung	Der Benutzer entfernt vom Hauptfenster die angezeigten Bilder.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden erfolgreich geladen.
Nachbedingung	Alle Anzeigefenster werden geleert und alle veränderten Einstellungen werden zurückgesetzt.
Basisablauf	Der Benutzer klickt auf den „Clear" Button, der mit dem Laden der Bilder aktiviert ist.
Alternative Abläufe	Drücken auf die „C" Taste.

5.4. Bilder erneut laden

Kurzbeschreibung	Der Benutzer lädt die vom Hauptfenster entfernten Bilder neu.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden vom Hauptfenster entfernt.
Nachbedingung	Die vom Hauptfenster entfernten Bilder werden erneut aufgerufen und angezeigt.
Basisablauf	Der Benutzer klickt auf den „Reload" Button, der mit dem Laden und der Entfernung der Bilder aktiviert ist.
Alternative Abläufe	Drücken auf die „R" Taste.

5.5. Bilder größer darstellen

Kurzbeschreibung	Der Benutzer stellt die im Hauptfenster angezeigten Bilder größer dar.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden erfolgreich eingelesen und dargestellt.
Nachbedingung	Die im Hauptfenster angezeigten Bilder werden im „Larger View" Fenster größer dargestellt.
Basisablauf	Der Benutzer klickt auf den „Movie" Button, der mit dem Laden der Bilder aktiviert ist.
Alternative Abläufe	Drücken auf die „M" Taste.
Anmerkung	Das „Larger View" Fenster kann auch genutzt werden, um schnell durch eine Serie von Bildern zu blättern.

5.6. Toolbereich öffnen/schliessen

Kurzbeschreibung	Der Benutzer macht den Toolbereich auf dem Hauptfenster sichtbar/unsichtbar.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden erfolgreich geladen.
Nachbedingung	Es wird rechts unten vom Hauptfenster ein Bereich geöffnet, der unterschiedliche Buttons und die Combobox enthält, um geeignete Änderungen über die dargestellte Bilder ermöglichen zu können
Basisablauf	Der Benutzer klickt auf den „Tool" Button, der mit dem Laden der Bilder aktiviert ist.
Alternative Abläufe	Drücken auf die „T" Taste.

5.7. Bilder segmentieren

Kurzbeschreibung	Der Benutzer segmentiert die geladenen 2D-Bilder.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden erfolgreich geladen.
Nachbedingung	Es wird ein 3D-Modell (Bild) erzeugt, das durch die eingelesenen 2D-Bilder und durch die Änderung der Grauwerte entsteht.
Basisablauf	Der Benutzer klickt auf den „3D" Button, der mit dem Laden der Bilder aktiviert ist.
Alternative Abläufe	Drücken auf die „3" Taste.

5.8. 3D-Modell speichern

Kurzbeschreibung	Der Benutzer speichert das 3D-Modell als vtkData.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden erfolgreich geladen und ein 3D-Modell wird dargestellt.
Nachbedingung	Das 3D-Modell wird als vtkData mit .txt Endung in einen gewünschten Ort/Ordner gespeichert.
Basisablauf	Der Benutzer klickt auf den „Save" Button, der mit der Segmentierung des 3D-Modells aktiviert ist. Es öffnet sich ein Chooser. Benutzer gibt den Dateiname ein und klickt auf den „OK" Button.
Alternative Abläufe	Drücken auf die „S" Taste.

5.9. Hilfefenster öffnen

Kurzbeschreibung	Der Benutzer öffnet das Hilfefenster.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Das Programm ist bereit gestartet.
Nachbedingung	Das Benutzerhandbuch wird im „Help View" Fenster bereit gestellt.
Basisablauf	Der Benutzer klickt auf den „Help" Button.
Alternative Abläufe	Drücken auf die „H" Taste.

5.10. Bilder blättern

Kurzbeschreibung	Der Benutzer blättert die dargestellten Bilder durch.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder sind geladen.
Nachbedingung	Durch Blättern eines Bildes wird die Bildansicht geändert.
Basisablauf	Der Benutzer hält mit der linken Maustaste den unter dem Anzeigefenster liegenden „Value Slider" und bewegt ihn zwischen seinen Grenzen (z.B. von 0 bis 70).
Alternative Abläufe	Statt Mausbewegung können linke und/oder rechte Richtungstasten benutzt werden.

5.11. Opazität ändern

Kurzbeschreibung	Der Benutzer ändert die Opazität der dargestellten Bilder.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Der Toolbereich ist geöffnet und unter der Combobox „Opacity“ wird irgendeine Ansicht ausgewählt.
Nachbedingung	Durch die Änderung des Opazitätswertes enthält die Ansicht unterschiedliche Undurchsichtigkeiten.
Basisablauf	Der Benutzer wählt unter der Combobox „Opacity“ irgendeine Ansicht und bewegt den unter der Combobox liegenden „Value Slider“.
Alternative Abläufe	Statt Mausbewegung können linke und/oder rechte Richtungstasten benutzt werden.

5.12. Bildfarbe ändern

Kurzbeschreibung	Der Benutzer ändert die Farbe der dargestellten Bilder.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Der Toolbereich ist geöffnet und unter dem Menü „Color“ wird irgendeine Ansicht und Farbe ausgewählt.
Nachbedingung	Durch die Änderung der Farben erhalten die Bilder gewünschte Farben.
Basisablauf	Der Benutzer wählt unter dem Menü „Color“ irgendeine Ansicht und Farbe aus.

5.13. Bildschirmfoto speichern

Kurzbeschreibung	Der Benutzer speichert ein Bildschirmfoto von irgendeiner Ansicht.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Der Toolbereich ist geöffnet und unter der Combobox „Screenshot" wird irgendeine Ansicht ausgewählt oder das „Larger View" Fenster ist geöffnet und der „Screenshot" Button ist aktiv.
Nachbedingung	Durch die Speicherung irgendeiner Ansicht wird ein .jpg Bild 300x300 oder 500x500 Pixel erhalten.
Basisablauf	Der Benutzer wählt unter der Combobox „Screenshot" irgendeine Ansicht und gibt beim geöffneten Chooser einen Dateiname ein und drückt auf den „OK" Button.

5.14. Bilder automatisch blättern

Kurzbeschreibung	Der Benutzer blättert irgendeine Ansicht automatisch (Filmvorführung / Animation).
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Das „Larger View" Fenster ist geöffnet.
Nachbedingung	Es wird eine Serie von irgendeiner Ansicht durch geblättert.
Basisablauf	Der Benutzer wählt im „Larger View" Fenster unter „Change View" irgendeine Ansicht und benutzt die unterschiedlichen Buttons (vorwärts, rückwärts, repeat usw.), die für das automatische Blättern geeignet ist.

5.15. Automatisches Blättern stoppen

Kurzbeschreibung	Der Benutzer stoppt das automatischen Blättern.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Das „Larger View“ Fenster ist geöffnet und automatisches Blättern von irgendeinem Ansicht ist gestartet.
Nachbedingung	Das Blättern wird angehalten und zum ersten Bild der Ansicht gewechselt.
Basisablauf	Der Benutzer drückt im „Larger View“ Fenster auf den „Stop“ Button, der mit einem „■“ Symbol gekennzeichnet ist.

5.16. Geschwindigkeit ändern

Kurzbeschreibung	Der Benutzer ändert die Geschwindigkeit der Animation
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Das „Larger View“ Fenster ist geöffnet und/oder automatisches Blättern von irgendeiner Ansicht ist gestartet.
Nachbedingung	Die Zwischenzeit des Blätterns ist geändert. Die Animation wird langsamer oder schneller abgespielt.
Basisablauf	Der Benutzer wählt unter der Combobox „Speed“ irgendeine Geschwindigkeitskategorie und wechselt zu denen Geschwindigkeitsgrad.

5.17. Grauwert ändern

Kurzbeschreibung	Der Benutzer ändert den Grauwert der dargestellten 2D-Bilder.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Die Bilder werden geladen und/oder „Larger View" Fenster ist geöffnet.
Nachbedingung	Der Grauwert eines Bildes bei den Ansichten („Axial", „Sagittal", „Coronal") ist nach Wahl für „Level" von -1000 bis 3000 und für „Width" von 0 bis 4096 geändert
Basisablauf	Der Benutzer führt den Mauszeiger auf irgendein Anzeigefenster („Axial", „Sagittal", „Coronal") und drückt auf die linke Maustaste, bewegt die Maus gedrückt haltend nach rechts, links, hinten und vorn.
Anmerkung	Die Grauwertänderungen sind nur auf den Anzeigefenstern „Display Axial", „Display Sagittal" und „Display Coronal" erlaubt.

5.18. Bilder drehen

Kurzbeschreibung	Der Benutzer dreht irgendeine Ansicht.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Die Bilder werden geladen und/oder „Larger View" Fenster ist geöffnet.
Nachbedingung	Der Sichtwinkel einer Ansicht (Bild) ist geändert.
Basisablauf	Der Benutzer führt den Mauszeiger auf irgendein Anzeigefenster („Display CAS Plane" oder „Display 3D") und drückt auf die linke Maustaste, bewegt die Maus gedrückt haltend nach rechts, links, hinten und vorn.
Anmerkung	Die Umdrehungen sind nur auf den Anzeigefenstern „Display CAS Plane" und „Display 3D" erlaubt.

5.19. Bildgröße ändern

Kurzbeschreibung	Der Benutzer ändert die Größe eines dargestellten Bildes.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Die Bilder werden geladen und/oder „Larger View" Fenster ist geöffnet.
Nachbedingung	Bilder werden auf den Anzeigefenstern kleiner oder größer dargestellt.
Basisablauf	Der Benutzer führt den Mauszeiger auf irgendein Anzeigefenster („Display Axial", „Display Sagittal", „Display Coronal", „Display CAS Plane", „Display 3D" oder „Larger View") und drückt auf die rechte Maustaste, bewegt die Maus gedrückt haltend nach vorn und hinten.

5.20. Bilder positionieren

Kurzbeschreibung	Der Benutzer ändert die Position eines dargestellten Bildes.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Die Bilder werden geladen und/oder „Larger View" Fenster ist geöffnet.
Nachbedingung	Die Standposition eines dargestellten Bildes wird geändert.
Basisablauf	Der Benutzer führt den Mauszeiger auf irgendein Anzeigefenster („Display Axial", „Display Sagittal", „Display Coronal", „Display CAS Plane", „Display 3D" oder „Larger View") und drückt auf die mittlere Maustaste, bewegt die Maus gedrückt haltend nach rechts, links, hinten und vorn.

5.21. Punkt selektieren

Kurzbeschreibung	Der Benutzer selektiert bei irgendeiner Ansicht („Axial", „Sagittal", „Coronal") einen Punkt/Bereich und alle anderen Ansichten zeigen dann denselben Punkt/Bereich.
Akteure/Auslöser	Benutzer/Benutzer
Vorbedingung	Bilder werden erfolgreich geladen.
Nachbedingung	Die Ansichten und Sichtwinkel aller dargestellten 2D-Bilder werden zur gleichen Ansicht und Sichtwinkel gebracht.
Basisablauf	Der Benutzer führt den Mauszeiger auf irgendein Anzeigefenster („Display Axial", „Display Sagittal", „Display Coronal") und drückt auf die linke Maustaste.
Anmerkung	Die Punktselektion ist nur bei den Ansichten „Display Axial", „Display Sagittal" und „Display Coronal" erlaubt.

6. Klassendiagramm

In diesem Abschnitt wird das Klassendiagramm von CTTranslator mit textuellen Beschreibungen und mit einem Klassendiagramm vorgestellt. Die Klassendiagramme sind eine graphische Darstellungsart, um die Programmklassen und Objekte eines Bereiches, die Eigenschaften (Attribute, Methoden, usw.), das Verhalten (Operationen) und die Beziehungen zwischen den Klassen deutlich zu modellieren[20].

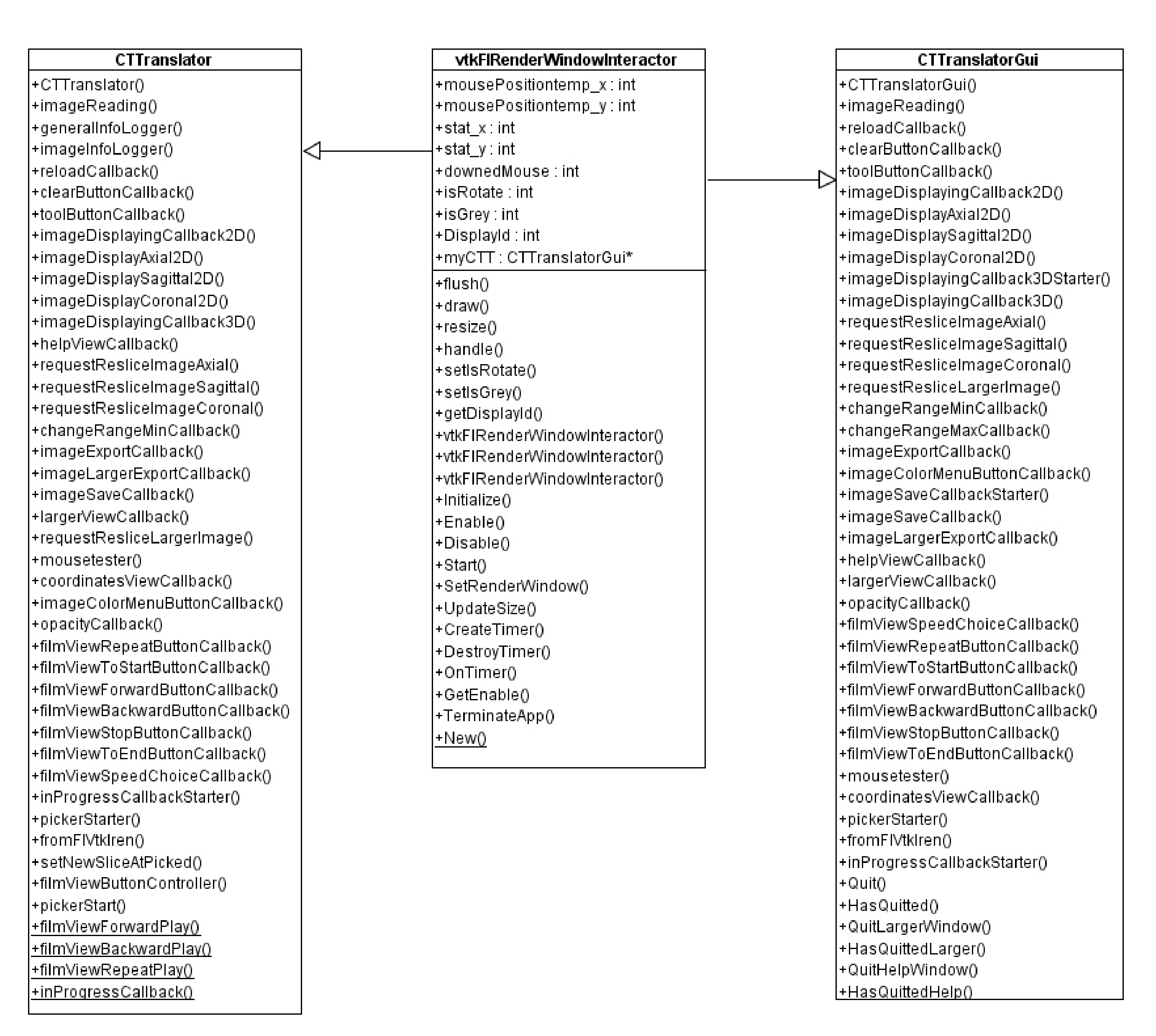


Abbildung 13: Klassendiagramm

6.1. CTTranslator

Die Klasse CTTranslator ist zuständig für alle Operationen, die oben im Klassendiagramm dargestellt sind. Die Klasse bezieht die Daten (Methodendefinitionen, FLTK-Widgets, usw.), die in der Klasse CTTranslatorGui vordefiniert sind, über die Klasse CTTranslatorGui. CTTranslator implementiert das Benutzerinterface, nimmt alle Benutzereingaben, transformiert Ereignisse in Aufrufe der Klasse vtkFlRenderWindowInteractor oder in Anfragen an die Klasse CTTranslatorGui. Die Eingaben der Benutzer werden von der Klasse CTTranslator verwaltet und auf die entsprechende Methoden der Klassen CTTranslatorGui oder vtkFlRenderWindowInteractor abgebildet. Bei der Instanziierung wird die Klasse CTTranslator von der Klasse CTTranslatorGui geerbt und die Definitionen der Funktionen werden in der Klasse CTTranslator implementiert.

6.2. CTTranslatorGui

Die Klasse CTTranslatorGui stellt für den Benutzer alle Benutzeroberflächenkomponenten (Buttons, Boxes, Slider, Input/Output-Felder, Tastatureingaben usw.) und Methodendefinitionen dar, die die Klasse CTTranslator von der Klasse CTTranslatorGui erbt.

6.3. vtkFlRenderWindowInteractor

Die Klasse vtkFlRenderWindowInteractor ist zuständig für alle Mausoptionen/Mausereignisse (Umdrehen, Positionieren, Vergrößern/Verkleinern, Koordinaten usw.), die der Benutzer für das Programm braucht.

7. Testdrehbuch

In diesem Abschnitt wird eine Anzahl von Testfällen beschrieben, welche durch unterschiedliche Bilddarstellung im alltäglichen Programmablauf auftreten werden/können.

Die Testfälle können im Groben unter folgende Kategorien unterteilt werden:

1. Bilder als 2D einlesen und darstellen.
2. Bilder betrachten und verarbeiten (geeignete Änderungen an den Bildern vornehmen).
3. Bilder als 3D darstellen (Segmentation).
4. Filmvorführung (Animation) durchführen.
5. 3D-Modell speichern.

Es wurden so viele Testfälle wie möglich abgehandelt, aber es versteht sich, dass hier nur die wichtigsten und am häufigsten auftretenden Fälle betrachtet werden können.

7.1. Referenzierte Dokumente

- Pflichtenheft (Use-Case Diagramm)
- GUI Prototyp

7.2. Beschreibung der Testfälle

Testfall: Bilder werden geladen und angezeigt (01)	
Name	Bilder laden und zeigen.
Use Case	Bilder laden.
Ausgangszustand	Programm ist bereit gestartet und „Load" Button ist aktiv.
Aktion	Der Benutzer startet das Programm, klickt auf den „Load" Button oder drückt auf die „L" Taste, wählt einen Ordner, in dem sich CT-Bilder befinden, aus und liest sie ein. Nachdem der Benutzer einen Ordner ausgewählt hat, wird das Laden von Bildern mit Klick auf den „OK" Button abgeschlossen.
erwarteter Ergebniszustand	Im Hauptfenster werden die eingelesenen Bilder in den Anzeigefenstern „Display Axial", „Display Sagittal", „Display Coronal" und „Display CAS Plane" als „Axial", „Sagittal", „Coronal" und „Planer" Ansichten als 2D dargestellt. Die Buttons „Reload", „Clear", „3D", „Movie", "Tool", "Help" und unter den Anzeigefenstern liegende Slider werden aktiviert. Auf dem Informationsfeld steht: „Images are successfully read and loaded" und „ Images are successfully displayed as 2D".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bilder werden entfernt (02)	
Name	Bilder entfernen.
Use Case	Bilder entfernen.
Ausgangszustand	Bilder sind erfolgreich geladen.
Aktion	Die Entfernung von Bildern und Zurücksetzung aller veränderten Einstellungen wird mit Klick auf den „Clear" Button oder Druck auf die „C" Taste abgeschlossen.
erwarteter Ergebniszustand	Es werden im Hauptfenster alle Ansichten entfernt und alle Einstellungen und Änderungen werden zurückgesetzt. Der Benutzer kann wieder mit dem „Load" Button neue Bilder lesen oder mit dem „Reload" Button zuletzt gelesene und angezeigte Bilder wieder aufrufen. Die Buttons: „Reload", „3D" und „Tool" werden wieder deaktiviert. Auf dem Informationsfeld steht: „All views and functions are successfully cleared! ".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bilder werden neu geladen (03)	
Name	Bilder neu laden.
Use Case	Bilder erneut laden.
Ausgangszustand	Bilder sind geladen oder vom Hauptfenster entfernt.
Aktion	Nachdem der Benutzer die Bilder erfolgreich geladen und/ oder die Bilder vom Hauptfenster entfernt hat, wird das Neuladen von Bildern mit dem Button „Reload" oder mit dem Druck auf die „R" Taste erzielt.
erwarteter Ergebniszustand	Der Benutzer erhält im Hauptfenster wieder zuletzt eingelesene Bilder (Es werden alle Darstellungen „Axial", „Sagittal", „Coronal", „CAS Plane", „3D" wieder aufgerufen.), die der Benutzer mit denen geeignete Veränderungen und Einstellungen übernehmen kann. Bei der neuen Ladung werden alle zuletzt veränderten Einstellungen zurückgesetzt. Die Buttons: „Clear", „3D", „Movie" und „Tool" werden wieder aktiviert. Auf dem Informationsfeld steht: „Images are successfully reloaded!"
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bilder werden größer dargestellt (04)	
Name	Bilder größer darstellen.
Use Case	Bilder größer darstellen.
Ausgangszustand	Bilder sind geladen und Button „Movie“ ist aktiv.
Aktion	Nachdem der Benutzer die Bilder erfolgreich geladen hat, klickt auf dem „Movie“ Button oder drückt auf die „M“ Taste.
erwarteter Ergebniszustand	Es wird ein neues Fenster geöffnet, auf dem am Anfang nur die „Axial“ Ansicht dargestellt ist. Der Benutzer kann auf diesem Fenster nach Wahl zu anderen Ansichten wechseln. Der Benutzer enthält auf dem „LargerView“ Fenster größere Ansichten von gleichen Bildern, die im Hauptfenster dargestellt sind. Auf dem „LargerView“ können wieder alle Veränderungen von einer Ansicht und vorgenommene Einstellungen übernommen werden. Auf dem Informationsfeld steht: „Image is successfully larger displayed!“.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Toolbereich wird geöffnet/geschlossen (05)	
Name	Toolbereich öffnen/schliessen.
Use Case	Toolbereich öffnen.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen.
Aktion	Der Benutzer klickt auf dem „Tool" Button oder drückt auf die „T" Taste.
erwarteter Ergebniszustand	Es wird ein Bereich im Hauptfenster sichtbar gemacht, auf dem sich Comboboxen „Screenschot", „Opacity" und „Color" befinden. um manche Änderungen der geladenen Bilder vornehmen zu können. Es ist jetzt, möglich Änderungen für angezeigte Bilder über diese Combobox durchzuführen und die Ergebnisse auf den angezeigten Bildern zu sehen. Auf dem Informationsfeld steht: „Properties area is successfully opened!" oder „Properties area is successfully closed! ".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bilder werden segmentiert (06)	
Name	Segmentierung der geladenen Bilder als 3D-Modell.
Use Case	Bilder segmentieren.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen.
Aktion	Der Benutzer klickt auf dem „3D" Button oder drückt auf die „3" Taste.
erwarteter Ergebniszustand	Es erscheint auf dem Anzeigefenster „Display 3D" ein 3D-Modell, das durch 2D-Bildern und Änderungen der Grauwerte entsteht. Auf dem Informationsfeld steht: „Images are successfully displayed as 3D! ".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: 3D-Modell wird als vtkData gespeichert (07)	
Name	3D-Modell als vtkData speichern.
Use Case	3D-Modell speichern.
Ausgangszustand	2D-Bilder werden segmentiert und der Button „Save" ist aktiv.
Aktion	Der Benutzer klickt auf dem „Save" Button oder drückt auf die „S" Taste.
erwarteter Ergebniszustand	Es wird ein Fenster „File Chooser" geöffnet. Die Speicherung ist mit der Eingabe des Dateinamens und Druck auf dem „OK" Button abgeschlossen. Auf dem Informationsfeld steht: „ 3D image is successfully saved as vtkData! ".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Hilfefenster wird geöffnet (08)	
Name	Hilfefenster öffnen.
Use Case	Hilfefenster öffnen.
Ausgangszustand	Programm ist gestartet.
Aktion	Der Benutzer klickt auf dem „Help" Button oder drückt auf die „H" Taste.
erwarteter Ergebniszustand	Es wird ein Fenster geöffnet, in dem sich alle Informationen und Hinweise über die Benutzung des Programmes befinden. Auf dem Informationsfeld steht: „ Help View is successfully displayed! ".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bildansicht wird geblättert (09)	
Name	Bilder blättern.
Use Case	Bilder blättern.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und/oder „Larger View" Fenster wird geöffnet.
Aktion	Der Benutzer bewegt mit linker Maustaste oder mit rechten/linken Richtungstasten unter den dargestellten Bildern liegenden „Value Slider".
erwarteter Ergebniszustand	Die Ansichten von den dargestellten Bildern werden per Bewegung von „Value Slider" vorwärts oder rückwärts geblättert und jedes Mal wird eine andere Ansicht, von den gleichen Bildern, die vom ausgewählten Ordner eingelesen wurde, dargestellt und angezeigt. Die Werte von „Value Slider" liegen für „Axial" zwischen 0 und den Anzahl der Bilder, die sich im eingelesenen Ordner befinden, für „Sagittal": zwischen 0 und den Bildgrößen (Bildauflösung z.B. 512), für „Coronal": zwischen 0 und den Bildgrößen.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Opazität wird geändert (10)	
Name	Opazität ändern.
Use Case	Opazität ändern.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und der Toolbereich ist geöffnet/sichtbar.
Aktion	Der Benutzer klickt auf „Tool" und wählt bei der Combobox „Opacity" irgendeine Ansicht (z.B. Axial) aus.
erwarteter Ergebniszustand	Nachdem der Benutzer eine Ansicht bei der Combobox „Opacity" ausgewählt hat, wird der darunter liegende „Value Slider" aktiviert. Der Benutzer kann nun den Slider zwischen den Werten 0.2 – 1.0 bewegen und die Opazität von dem ausgewählten Bild ändern.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bildfarbe wird geändert (11)	
Name	Bildfarbe ändern.
Use Case	Bildfarbe ändern.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und der Toolbereich ist geöffnet/sichtbar.
Aktion	Der Benutzer klickt auf „Color" und wählt aus der Liste die gewünschte Ansicht und Farbe aus.
erwarteter Ergebniszustand	Nachdem der Benutzer eine Ansicht und Farbe im Menü „Color" ausgewählt hat, wird das angezeigte Bild in die gewünschten Farbe gefärbt.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bildschirmfoto wird gespeichert (12)	
Name	Bildschirmfoto speichern.
Use Case	Bildschirmfoto speichern.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und der Toolbereich ist sichtbar und/oder das „Larger View" Fenster ist geöffnet.
Aktion	Der Benutzer klickt auf „Tool" und wählt bei der Combobox „Screenshot" irgendeine Ansicht (z.B. Axial) aus oder der Benutzer klickt im „Larger View" Fenster auf den „Screenshot" Button.
erwarteter Ergebniszustand	Nachdem der Benutzer im Hauptfenster eine Ansicht bei der Combobox „Screenshot" ausgewählt hat oder im „Larger View" Fenster auf dem „Screenshot" Button geklickt hat, wird ein Fenster „File Chooser" geöffnet. Die Speicherung des Bildschirmfotos ist mit der Wahl des Speicherortes (z.B. in einen Ordner), in dem das Bildschirmfoto gespeichert werden soll, mit der Eingabe des Dateinamens und Druck auf dem „OK" Button abgeschlossen. Auf dem Informationsfeld steht: „Image is successfully exported as .jpg (300x300)! "oder „Image is successfully exported as .jpg(500x500)!".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Filmvorführung wird gestartet (vorwärts) (13)	
Name	Filmvorführung (Animation) vorwärts starten.
Use Case	Bilder automatisch blättern.
Ausgangszustand	„Larger View“ Fenster ist geöffnet.
Aktion	Der Benutzer klickt auf „vorwärts“ Button, der mit dem Symbol „>“ gekennzeichnet ist.
erwarteter Ergebniszustand	Der Wert von „Value Slider“ wird gleich auf den Wert 0 (erstes Bild) angebracht und es wird gleich angefangen jedes Bild vorwärts zu zeigen. Bei der Vorführung des x-ten Bildes bewegt sich der Slider immer vorwärts und dabei wird sein Wert um 1 erhöht. Der Slider geht bis zum Ende seines Wertes. Wenn der Slider bis zum Ende gekommen ist (alle Bilder werden in gleicher Ansicht angezeigt. z.B. „Axial“), wird die Animation angehalten. Beim Lauf der Animation werden: die Combobox „Change View“, die Buttons „repeat“, „rückwärts“, „vorwärts“ und „Screenshot“ deaktiviert. Auf dem Informationsfeld steht: „ Filmview is successfully launched!“.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Filmvorführung wird gestartet (rückwärts) (14)	
Name	Filmvorführung (Animation) rückwärts starten.
Use Case	Bilder automatisch blättern.
Ausgangszustand	„Larger View“ Fenster ist geöffnet.
Aktion	Der Benutzer klickt auf „rückwärts“ Button, der mit dem Symbol „<“ gekennzeichnet ist.
erwarteter Ergebniszustand	Der „Value Slider“ wird gleich bis zum Ende seines Wertes (letztes Bild) angebracht und es wird gleich angefangen jedes Bild zu zeigen. Bei der Vorführung des x-ten Bildes bewegt sich der Slider immer ein Wert rückwärts, dabei wird sein Wert um 1 gesenkt. Der Slider geht bis zum Anfang des Bildes (auf Wert 0). Wenn der Slider bis zum Anfang gekommen ist, wird die Animation angehalten. Beim Lauf der Filmansicht werden: die Combobox „Change View“, die Buttons „repeat“, „rückwärts“, „vorwärts“ und „Screenshot“ deaktiviert. Auf dem Informationsfeld steht: „ Filmview is successfully launched!“.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Filmvorführung wird gestartet (wiederholtes Blättern) (15)	
Name	Filmvorführung (Animation) vorwärts und rückwärts starten.
Use Case	Bilder automatisch blättern.
Ausgangszustand	„Larger View" Fenster ist geöffnet.
Aktion	Der Benutzer klickt auf „repeat" Button, der mit dem Symbol „<->" gekennzeichnet ist.
erwarteter Ergebniszustand	Der „Value Slider" wird gleich auf den Wert 0 (erstes Bild) angebracht und es wird gleich angefangen jedes Schnittbild vorwärts zu zeigen. Bei der Vorführung des x-ten Bildes bewegt sich der Slider immer vorwärts und dabei wird sein Wert um 1 erhöht. Der Slider geht bis zum Ende seines Wertes. Wenn der Slider bis zum Ende gekommen ist, (alle Bilder werden in gleicher Ansicht angezeigt. z.B. „Axial"), werden die gleiche Bilder im angegebenen Zeitintervall rückwärts angezeigt, dabei bewegt sich der Sleider rückwärts und sein Wert wird um 1 gesenkt (endlose Schleife). Beim Lauf der „repeat" Animation werden: die Combobox „Change View", die Buttons „repeat", „rückwärts", „vorwärts" und „Screenshot" deaktiviert. Auf dem Informationsfeld steht: „Filmview is successfully launched!".
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

<table>
<tr><td colspan="2">Testfall: Geschwindigkeit wird geändert (16)</td></tr>
<tr><td>Name</td><td>Animationsgeschwindigkeit ändern.</td></tr>
<tr><td>Use Case</td><td>Geschwindigkeit ändern.</td></tr>
<tr><td>Ausgangszustand</td><td>„Larger View“ Fenster ist geöffnet oder es wird eine Filmvorführung von irgendeiner Ansicht („Axial“, „Sagittal“ oder „Coronal“) in Lauf gesetzt.</td></tr>
<tr><td>Aktion</td><td>Der Benutzer wählt aus der Combobox irgendeinen Geschwindigkeitsregler („Low“, „Normal“, „High“).</td></tr>
<tr><td>erwarteter Ergebniszustand</td><td>Es wird die Laufgeschwindigkeit der Filmvorführung geändert. Somit werden die Zeitintervalle bei der Wartezeit der Darstellung von Bildern kürzer oder länger eingestellt. Die Wartezeit bei der Darstellung für die Geschwindigkeitskategorien betragen: „Low: 5 ms“, „Normal: 3 ms“, „High: 1 ms“.</td></tr>
<tr><td colspan="2">beobachtete Abweichung</td></tr>
<tr><td colspan="2">☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar</td></tr>
</table>

Testfall: Filmvorführung wird gestoppt (17)	
Name	Animation stoppen.
Use Case	Benutzer stoppt die Animation.
Ausgangszustand	„Larger View" Fenster ist geöffnet und es wird eine Filmvorführung von irgendeiner Ansicht („Axial", „Sagittal" oder „Coronal") in Lauf gesetzt.
Aktion	Der Benutzer klickt auf den „stop" Button, der mit dem Symbol „ ■" gekennzeichnet ist.
erwarteter Ergebniszustand	Die Filmvorführung wird angehalten und der Slider wird auf den Wert 0 gebracht. Nach dem Abstellung der Filmvorführung werden: die Combobox „Change View", die Buttons „vorwärts und rückwärts", „rückwärts", „vorwärts" und „Screenshot" wieder aktiviert.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Grauwert wird geändert (18)	
Name	Grauwert ändern.
Use Case	Grauwert ändern.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und/oder das „Larger View" Fenster ist geöffnet.
Aktion	Der Benutzer klickt im Hauptfenster oder im „Larger View" Fenster mit linker Maustaste auf angezeigtes Bild, während der Mauszeiger auf einer Bildansicht steht (z.B. „Axial"). Und bewegt die Maus gedrückt haltend nach rechts, links, hinten und vorn.
erwarteter Ergebniszustand	Die Bilder werden am Anfang zwischen den Grauwerten (-250 bis 1200) gelesen und dargestellt. Wenn der Benutzer den Grauwert ändert, wird das Ergebnis auf den Ansichten „Axial", „Sagittal", „Coronal" und „CAS Plane" angezeigt. Auf dem „Progress bar" werden auch Koordinaten vom Mauszeiger und momentanes Grauwertintervall angezeigt. Die Änderung des Grauwertes erfolgt im Hauptfenster oder im „Larger View" Fenster nur in Ansichten „Axial", „Sagittal" und „Coronal". Die Beendigung der Grauwertänderung wird durch Freilassen der linken Maustaste abgeschlossen.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bilder werden gedreht (19)	
Name	Bilder drehen.
Use Case	Bilder drehen.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und/oder das „Larger View" Fenster wird geöffnet.
Aktion	Der Benutzer klickt im Hauptfenster oder im „Larger View" Fenster mit linker Maustaste auf angezeigtes Bild, während der Mauszeiger auf einem Anzeigefenster „CAS Plane" oder „Display 3D" steht. Und bewegt die Maus gedrückt haltend nach rechts, links, hinten und vorn.
erwarteter Ergebniszustand	Die Bildansichten werden im Rahmen der 3D-Koordinatenrichtungen gedreht und nach Wahl zu gewünschten Sichtwinkeln gebracht. Die Beendigung der Umdrehung wird durch Freilassen der linken Maustaste abgeschlossen.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bildgröße wird geändert (20)	
Name	Bildgröße ändern.
Use Case	Bildgröße ändern.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und/oder „Larger View" Fenster wird geöffnet.
Aktion	Der Benutzer klickt im Hauptfenster oder im „Larger View" Fenster mit rechter Maustaste auf angezeigtes Bild, während der Mauszeiger auf einem Anzeigefenster „Axial", „Sagittal", „Coronal", „CAS Plane" oder „Display 3D" steht. Und schiebt die Maus gedrückt haltend nach vorn und hinten.
erwarteter Ergebniszustand	Es wird eine Vergrößerung erzielt, wenn der Benutzer die Maus nach hinten bewegt/schiebt und eine Verkleinerung, wenn der Benutzer die Maus nach vorn schiebt. Die Beendigung der Vergrößerung oder Verkleinerung wird durch Freilassen der rechten Maustaste abgeschlossen.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Bilder werden positioniert (21)	
Name	Bilder positionieren.
Use Case	Bilder positionieren.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen und/oder „Larger View" Fenster wird geöffnet.
Aktion	Der Benutzer klickt im Hauptfenster oder im „Larger View" Fenster mit mittlerer Maustaste auf angezeigtes Bild, während der Mauszeiger auf irgendeinem Anzeigefenster „Axial", „Sagittal", „Coronal", „CAS Plane" oder „Display 3D" steht und bewegt die Maus gedrückt haltend nach rechts, links, hinten und vorn oder der Benutzer bestimmt im Anzeigefenster einen Bereich/Punkt, führt die Maus hin und klickt auf mittlere Maustaste. Das Bild wird dann auf diesem Bereich/Punkt positioniert.
erwarteter Ergebniszustand	Im Fenster wird eine Bewegung der Bilder erzielt, wenn der Benutzer die Maus nach rechts, links, hinten und vorn bewegt oder auf den vorher bestimmten Bereich klickt. Die Beendigung der Positionierung wird durch Freilassen der mittleren Maustaste abgeschlossen.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

Testfall: Punkt wird selektiert (22)	
Name	Punkt selektieren.
Use Case	Punkt selektieren.
Ausgangszustand	Das Laden oder Neuladen von Bildern ist erfolgreich abgeschlossen.
Aktion	Der Benutzer selektiert (klickt) im Hauptfenster mit linker Maustaste auf angezeigtes Bild, während der Mauszeiger auf irgendeinem Anzeigefenster „Axial", „Sagittal", „Coronal" steht.
erwarteter Ergebniszustand	Nachdem der Benutzer irgendeinen Punkt selektiert hat (z.B. linkes Auge bei der „Axial" Ansicht), wird der gleiche Punkt bei den „Sagittal" und „Coronal" Ansichten (seitlich und von vorne) angezeigt. Die Beendigung der Punktselektion wird durch Freilassen der linken Maustaste abgeschlossen.
beobachtete Abweichung	
☒ OK ☐ kosmetische Abweichungen ☐ mittlere Abweichungen ☐ große Abweichungen ☐ System unbenutzbar	

8. Benutzerhandbuch

8.1. Installation

Vor Installation des Programms müssen die VTK, FLTK und CMake Bibliotheken installiert werden (genauere Erklärung finden Sie unter LIESMICH.txt Datei auf der Installations-CD). Nachdem alle Grundvoraussetzungen am System erfüllt sind, kann der CTTranslator konfiguriert, kompiliert und gestartet werden.

8.2. Programm starten

Auf der Installations-CD befindet sich das Verzeichnis "CTTranslator". Kopieren Sie dieses von der CD in Ihr Homeverzeichnis. Starten sie unter Linux eine Konsole oder ein Terminal und wechseln Sie z.B. mit „cd /Desktop/CTTranslator/" in dieses Verzeichnis. Um das Programm zu starten geben sie „./CTTranslator" ein.

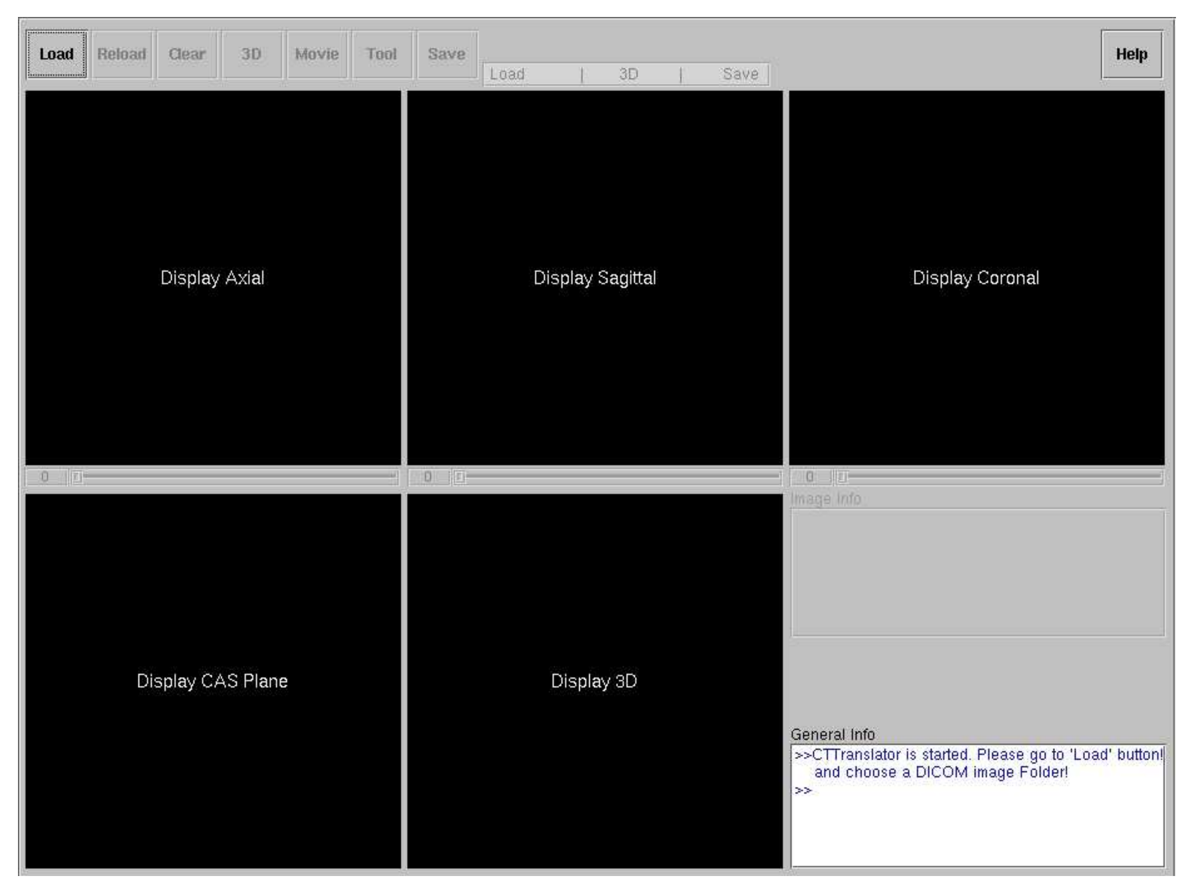


Abbildung 14: CTTranslator Hauptfenster

8.3. Bilder einlesen und anzeigen

Abbildung 15: 2D-Bilder

Load Nachdem Sie das Programm gestartet haben, gehen sie auf den „Load" Button und klicken Sie darauf oder drücken Sie auf die „L" Taste. In der Folge dieser Aktionen öffnet sich ein File Chooser. Wählen Sie dort einen Ordner aus, in dem sich nur DICOM/ .dcm formatierte Computer Tomographie (CT) Bilder befinden. Um das Laden der Bilder abzuschließen, klicken sie auf dem File Chooser Fenster auf „OK".

Hinweis: Das Programm akzeptiert nur CT-Bilder, die mit .dcm Endung formatiert und fortlaufend nummeriert sind. Die geladenen Bilder werden in den Anzeigefenster, „Display Axial", „Display Sagittal", „Display Coronal" und „Display CAS Plane" als 2D in den Ansichten „Axial", „Sagittal", „Coronal" und „Multiplanar" angezeigt. Die Buttons: „Reload", „Clear", „3D", „Movie", „Tool", die Progress Bar und alle Slider werden aktiviert.

8.4. Bilder entfernen

Clear Um auf dem Anzeigefenster dargestellte Bilder vom Hauptprogramm zu entfernen und alle vorher übernommenen Einstellungen oder Veränderungen zurück zu setzen, klicken Sie auf den „Clear" Button oder drücken Sie auf die „C" Taste.

Hinweis: Wenn das „Larger View" Fenster auch geöffnet ist, wird es durch den Klick auf den „Clear" Button geschlossen. Sie können gleiche Bilder wieder mit dem „Reload" Button aufrufen oder mit dem „Load" Button neue Bilder laden und anzeigen lassen. Nach dem Klick auf den „Reload" Button wird das geschlossene „Larger View" Fenster nicht geöffnet.

8.5. Bilder neu laden

Reload Das neue Laden von entfernten Bildern erfolgt durch den Klick auf den „Reload" Button oder Druck auf die „R" Taste. Nach dem neuen Laden werden alle übernommenen Einstellungen und Veränderungen zurückgesetzt.

8.6. Bilder größer darstellen

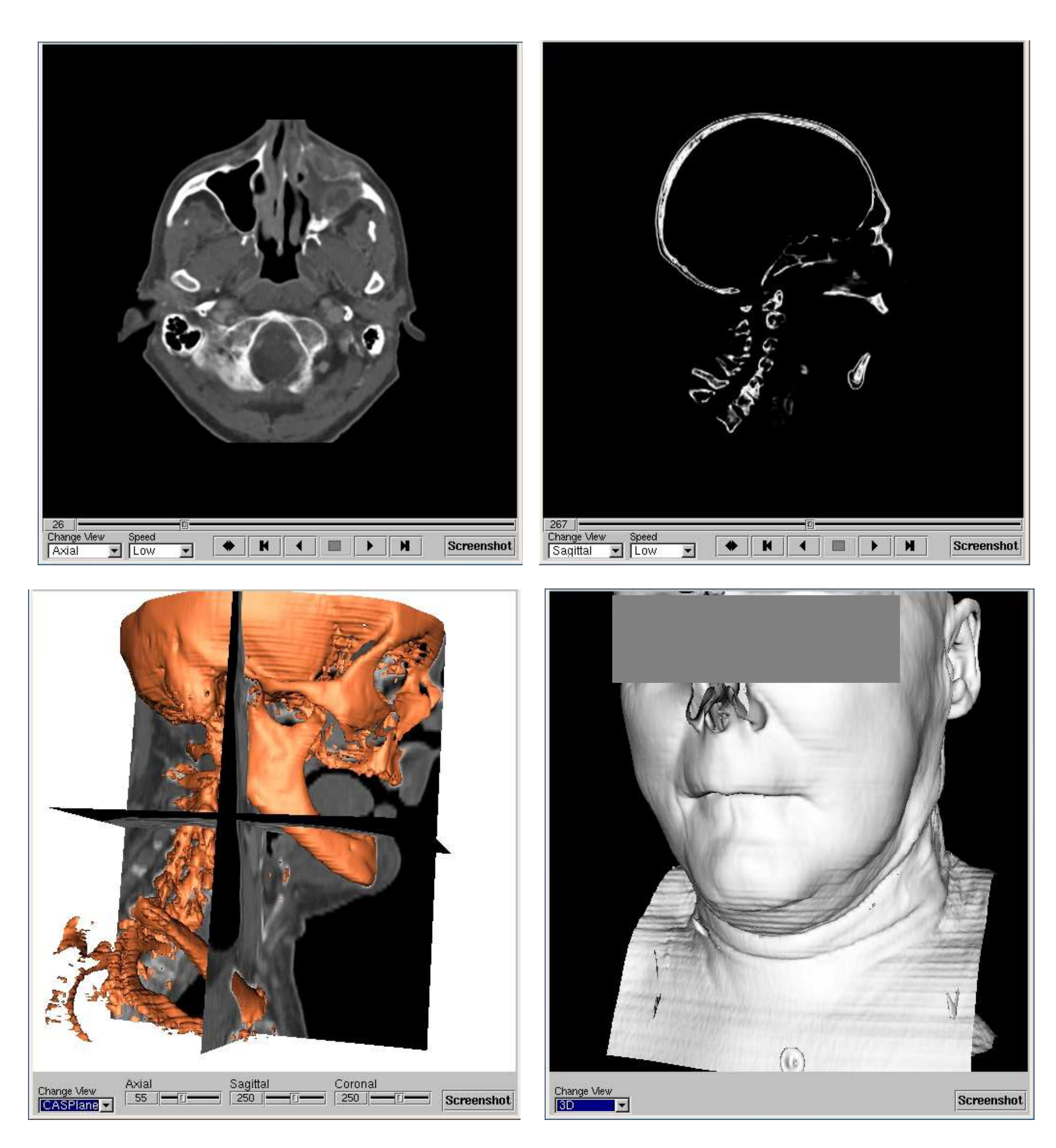


Abbildung 16: LargerView Fenster

Movie

Die im Hauptfenster angezeigten Bilder („Axial", „Sagittal", „Coronal", „CAS Plane" und „3D") können auch größer dargestellt werden. Dafür klicken Sie auf den „Movie" Button oder drücken Sie auf die „M" Taste. „Larger View" kann auch genutzt werden, um schnell durch eine Serie von Bildern zu blättern.

Es kann der im Hauptfenster oder im „Larger View" Fenster unter den Anzeigefenstern liegende Slider benutzt werden, um die angezeigten Bilder „Axial", „Sagittal" und „Coronal" durch die Bewegung der Slider vorwärts oder rückwärts manuell (per Hand) zu blättern.

Mit der Combobox „Change View" können im „Larger View" Fenster dargestellte Ansichten gewechselt werden.

Der „Automatischer vorwärts – rückwärts Vorführungsmodus (repeat)" Button ermöglicht das „vorwärts - rückwärts " wiederholende Blättern (endlose Schleife) der Bilder in gleicher Ansicht.

Der „Sprung zum Anfang" Button ermöglicht den Slider schnell zum Anfang zu führen.

Der „Automatischer rückwärts Vorführungsmodus" Button ermöglicht das Rückwärtsblättern der Bilder in gleicher Ansicht.

Der „Stop" Button ermöglicht die Filmvorführung anzuhalten.

Der „Automatischer vorwärts Vorführungsmodus" Button ermöglicht das Vorwärtsblättern der Bilder in gleicher Ansicht.

Der „Sprung zum Ende" Button ermöglicht den Slider schnell zum Ende zu führen.

Durch „Speed" Combobox kann bei der automatischen Vorführung/ Blättern der Bilder die Geschwindigkeit geregelt werden. Damit wird die Zwischenzeit beim Blättern kürzer oder länger eingestellt.

Screenshot Nach Öffnung des „Larger View" Fensters oder nach Wechseln zu irgendeiner Ansicht können durch den „Screenshot" Button die angezeigten Bilder als Bildschirmfotos gespeichert werden.

Hinweis: Wenn im Hauptfenster kein 3D-Bild dargestellt ist, bevor das „Larger View" Fenster geöffnet wird, ist in der Combobox „Change View" ein Wechsel zur 3D-Ansicht nicht möglich. Bei der automatischen Filmvorführung werden die Buttons: „wiederholtes Blättern", „Sprung zum Anfang", „rückwärts", „vorwärts" „Sprung zum Ende", „Screenshot" und die Combobox „Change View" deaktiviert. Nach dem anhalten der Vorführung werden sie wieder aktiviert.

8.7. Toolbereich öffnen

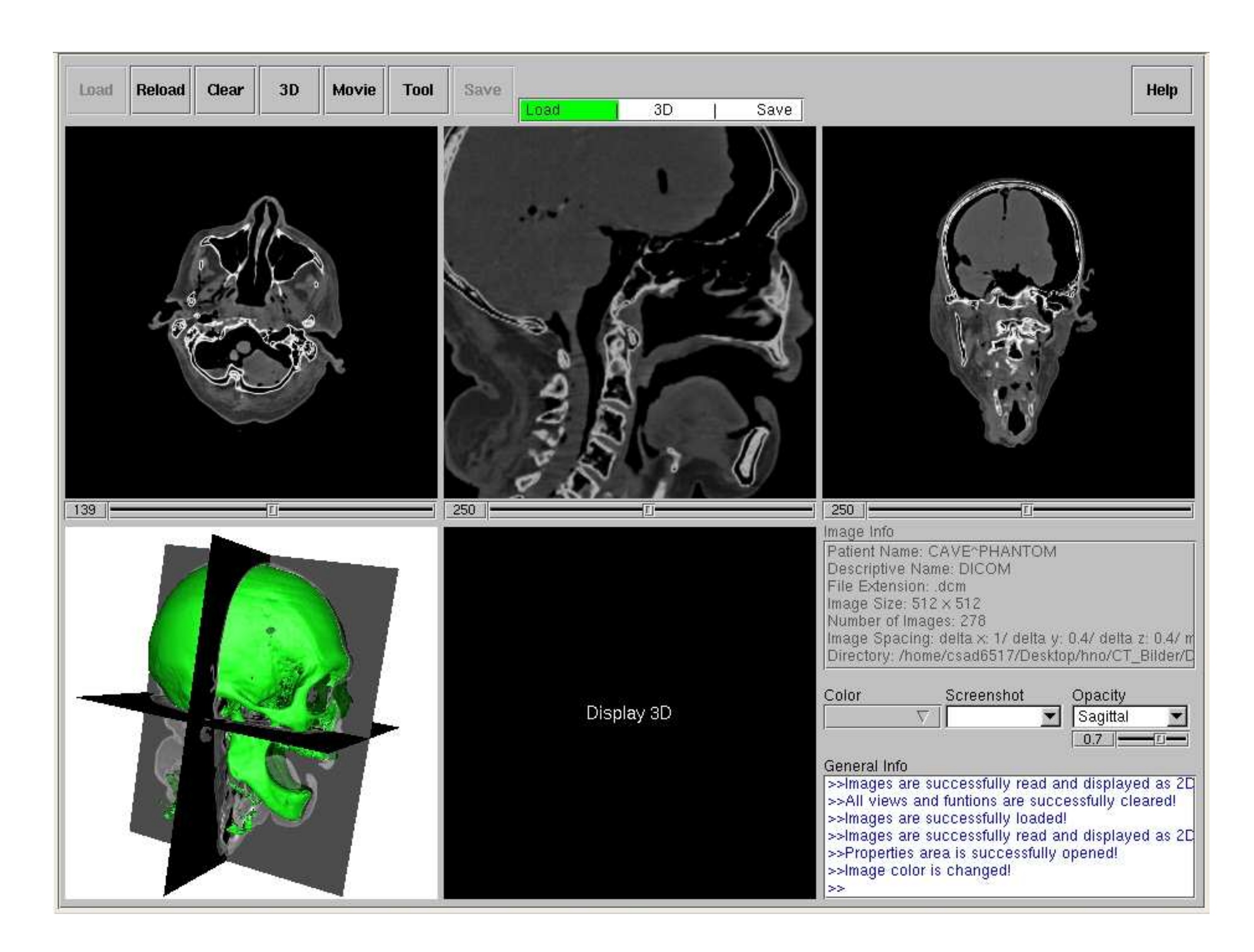


Abbildung 17: Hauptfenster und Toolbereich

Nachdem Bilder geladen oder neu geladen sind, kann der Toolbereich durch einen Klick auf den „Tool“ Button oder Druck auf die „T“ Taste geöffnet werden, um bestimmte Änderungen und Bearbeitungen übernehmen zu können.

Nach Öffnung des Toolbereichs kann durch die „Schreenshot“ Combobox irgendeine Ansicht der angezeigten Bildern ausgewählt und als Bildschirmfoto gespeichert werden.



Um die Opazität irgendeiner Ansicht zu ändern, wählen Sie in der Liste eine Ansicht und bewegen Sie die unten stehenden und nach dem Auswahl aktivierten Slider zwischen Werten 0.2 und 1.0.

Um die Farbe von irgendeiner Ansicht zu ändern, wählen Sie unter dem „Color“ Menü eine Ansicht und die gewünschte Farbe aus.

Hinweis: Wenn die Bilder durch „Clear“ im Hauptfenster entfernt sind, während der Toolbereich geöffnet war, wird der Bereich nach dem neuen Laden durch „Reload“ wieder nicht geöffnet.

8.8. Bilder segmentieren

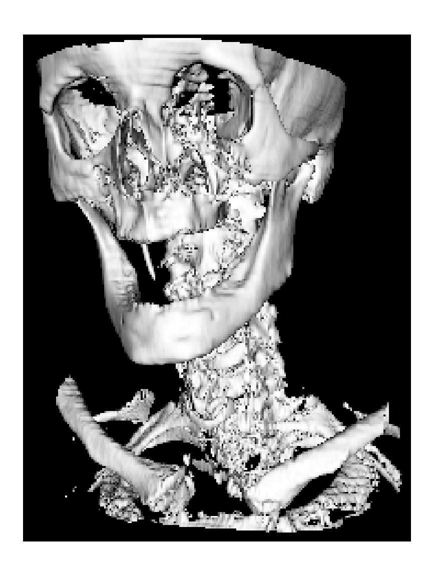
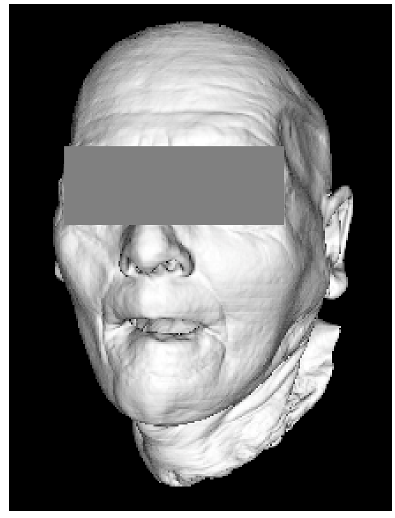



Abbildung 18: Knochenstruktur und Weichgewebe als 3D

3D Als 2D angezeigte Bilder können durch einen Klick auf den „3D" Button oder Druck auf die „3" Taste 3 Dimensional dargestellt werden. Die verschiedenen Ansichten (Knochenstruktur, Weichgewebe oder beide) eines 3D-Bildes erfolgt durch die Änderung des Grauwertes. Sie können gleich nach dem Darstellung der 2D-Bilder auf „3D" klicken und ein 3D-Bild mit vordefinierten Grauwerten zwischen -250 und 1200 im Anzeigefenster „Display 3D" darstellen lassen oder zuerst den Grauwert ändern und wieder auf „3D" klicken.

Bei der Phase „Bilder einlesen und anzeigen" wird im Anzeigefenster „CAS Plane" auch ein 3D-Modell zwischen Grauwerten -500 und 1000 auf 2D-Multiplanaransicht dargestellt.

i **Hinweis:** Wenn Sie in einem anderen Programm ein 3D-Modell darstellen und mit dem 3D-Modell von CTTranslator vergleichen möchten, berücksichtigen Sie die Scann-Richtung der Segmentierung und das Koordinatensystem der Bilder (Drei-Finger-Regel, Rechte-Hand-Regel oder Linke-Hand-Regel), ablesbar in der Headerdatei der CT-Bilder.

8.9. 3D-Modell speichern

Save

Ein als 3D dargestelltes Modell kann durch den Klick auf den „Save" Button oder Druck auf die „S" Taste als vtkData gespeichert werden. Nachdem Sie auf den „Save" Button geklickt haben, wird ein „File Chooser" geöffnet. Um die Speicherung der Datei abzuschließen, wählen Sie einen Ort/Ordner, in den die Datei gespeichert werden soll, geben Sie einen Name ein und klicken Sie auf den „OK" Button.

8.10. Hilfefenster öffnen



Um bestimmte Funktionsinformationen über das Programm erhalten zu können klicken Sie auf den „Help" Button oder drücken Sie auf die „H" Taste.

8.11. Allgemeine Informationsfenster



Alle Aktionen, die bei der Benutzung des Programms durchgeführt wurden, werden in dem „General Info" Fenster als Text protokolliert.

8.12. Bildinformationsfenster



Nach dem Laden der CT-Bilder werden in diesem Fenster Patient-Name, Bildformat, Dateiendung, Bildgröße, Bildanzahl, Abstände zwischen den Bildern und den Pfad, wo sich die Bilder befinden, aufgelistet.

8.13. Mausfunktionen

8.13.1. Linke Maustaste

Durch die Änderung des Grauwertes können unterschiedliche Ergebnisse von dargestellten Bildern erhalten werden. Um das zu ermöglichen, führen Sie den Mauszeiger auf irgendein Anzeigefenster „Axial", „Sagittal" oder „Coronal", drücken Sie auf die linke Maustaste und bewegen Sie die Maus gedrückt haltend nach rechts, links, hinten und vorn. Die Ergebnisse der Änderungen sind in den Bereichen „Display Axial", „Display Sagittal", „Display Coronal" und „Display CAS Plane" sichtbar.

Um das Bild zur gewünschten Position bringen zu können, kann eine Ansicht gedreht werden. Dafür führen Sie den Mauszeiger auf das Anzeigefenster „Display CAS Plane" oder auf „Display 3D", drücken sie auf die linke Maustaste und bewegen Sie die Maus gedrückt haltend nach rechts, links, hinten und vorn.

Um einen gewünschten Punkt in irgendeiner Ansicht (z.B. bei der „Axial" Ansicht) auch in anderen Ansichten („Sagittal" und „Coronal") sehen zu können, klicken Sie mit der linken Maustaste auf den gewünschten Bereich beim „Axial" Anzeigefenster. Somit sehen Sie den gleichen Punkt, den Sie bei der „Axial" Ansicht selektiert haben auch in den „Sagittal" und „Coronal" Ansichten (seitlich und von vorne).

X: 66 Y: 268 L: -288 W: 756 Bei der Verwendung der linken Maustaste werden auf dem „Progress Bar" aktuelle Koordinaten und Grauwertintervall angezeigt.

8.13.2. Rechte Maustaste

Die angezeigten Bilder können vergrößert oder verkleinert werden. Für die Vergrößerung drücken Sie auf die rechte Maustaste und bewegen Sie die Maus gedrückt haltend nach hinten. Für die Verkleinerung, bewegen Sie die Maus gedrückt haltend nach vorn.

8.13.3. Mittlere Maustaste

Die Position von einem dargestellten Bild kann im Rahmen des Anzeigefensters geändert/verschiebt werden. Während der Mauszeiger auf irgendeinem Anzeigefenster steht, drücken Sie auf die mittlere Maustaste und bewegen Sie die Maus gedrückt haltend rechts, links, hinten und vorne oder bestimmen Sie einen Ort im Anzeigefenster, führen Sie den Mauszeiger hin und drücken Sie auf die mittlere Maustaste.

Hinweis: Alle Mausoptionen, außer Punktselektion, können auch im „Larger View" Fenster angewendet werden.

8.14. Programm beenden

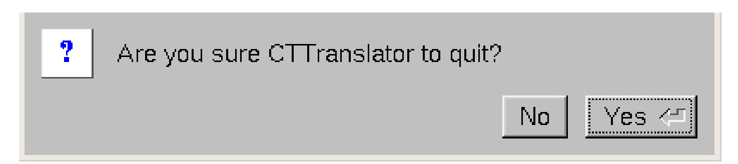


Abbildung 19: CTTranslator Beendigungsfenster

Wenn Sie das Programm verlassen möchten, klicken Sie im Hauptfenster oben rechts auf den „x" Button oder drücken auf die „Esc" Taste und klicken Sie im geöffneten Popup Fenester auf den „Yes" Button.

Hinweis: Wenn beim Verlassen des Programms die „Larger View" und „Help" Fesnter auch geöffnet sind, werden auch diese geschloßen.

9. Danksagung

Zum Schluss möchte ich mich bei Herrn Prof. Wolfgang Freysinger für die Überlassung des Themas und die freundliche Unterstützung und Betreuung der Arbeit bedanken.

Weiters möchte ich mich bei den Herren Dipl.-Ing. Özgür Güler und Abdullatif Yesil für ihre Betreuungen bedanken. Sie standen mir immer mit Rat und Verbesserungsvorschlägen zur Seite.

10. Anhang

10.1. Dateien

Das Programm besteht aus folgenden Dateien.

10.1.1. Ausführbare Dateien

CTTranslator

10.1.2. Programmdateien

CTTranslator.cpp

Enthält alle Ressourcen (Präprozessoren, Funktionen, Variablen, Instanzen, Zugriffskontrolle, Konstruktoren, Konstanten, Anweisungen, Schleifen, Objekte, Zeiger, usw.), die für das Programm nötig sind.

CTTranslator.h

Die Header-Datei von der Klasse CTTranslator.cpp enthält alle Deklarationen von inline und statische Funktionen, virtuelle Funktionendeklarationen, Präprozessoren, die in der Klasse CTTranslator.cpp benutzt und aufgerufen werden.

CTTranslatorGui.cpp

Enthält alle Ressourcen (Funktionen, Deklarationen usw.) die für die Benutzeroberfläche des Programms nötig sind.

CTTranslatorGui.h
Die Header-Datei von der Klasse CTTranslatorGui.cpp.

CTTranslatorGui.fl

Enthält alle Benutzeroberflächenelemente (Buttons, Outputfelder, Slider, Boxes usw.) des Programms.

vtkFlRenderWindowInteractor.cpp

Enthält alle Präprozessoren, Funktionen, Variablen, Konstruktoren, Konstanten, Anweisungen, Objekte, Zeiger, usw., die der Benutzter bei den Mausoptionen braucht (Steuerung für Bildpositionierung, Bildumdrehung, Änderung der Bildgröße usw.).

vtkFlRenderWindowInteractor.h

Die Header-Datei von der Klasse vtkFlRenderWindowInteractor.cpp

CmakeLists.txt

Enthält die Bibliotheknamen, und denen Pfade und die Dateinamen, die bei der Konfigurierung und Kompilierung des Programms nötig sind.

info.cpp

In dieser Klasse werden alle Informations- und Warnungsnachrichten (Konstanten) definiert, die im Gebrauch von der Klasse CTTranslator.cpp aufgerufen werden.

main.cpp

Die Hauptklasse.

10.1.3. Bibliothekdateien

iostream

FL/Fl_Progress.H

FL/Fl_File_Chooser.H

vtkFlRenderWindowInteractor.h

vtkMarchingCubes.h

vtkPointPicker.h

vtkPolyDataWriter.h

vtkPolyDataMapper.h

vtkImageToImageFilter.h

vtkImageMapToColors.h

vtkProperty.h

vtkRenderWindow.h

vtkRenderer.h

vtkDICOMImageReader.h

vtkActor.h

vtkImageActor.h

vtkLookupTable.h

vtkJPEGWriter.h

vtkCallbackCommand.h

vtkCamera.h

10.2. Abbildungsverzeichnis

11. Literaturverzeichnis

[1] Will Schroeder, Ken, Martin, Bill Lorensen, „*VTK User's Guide (VTK 4.4) Install, Use and Extend The Visualization ToolKit*", ISBN: 1930934130, (2005)

[2] Kitware, Inc., "*The Visualization ToolKit*", http://www.vtk.org/, (2008)

[3] Kitware, Inc., "*The Visualization ToolKit Class List*", http://www.vtk.org/doc/nightly/html/classes.html, (2008)

[4] Michael Bender, Manfred Brill, „*Computergrafik. Ein anwendungsorientiertes Lehrbuch 2. Auflage*", Hanser Wirtschaft ISBN: 3446404341 (2005)

[5] Fast Light ToolKit, http://www.fltk.org/, (2008)

[6] Michael Sweet, Craig P. Earls, and Bill Spitzak, "*FLTK 1.1.7 Programming Manual (Revision 7)*", By Bill Spitzak and Others (1998-2006)

[7] Xenophon Papademetris, "*An Introduction to Programming for Medical Image Analysis with the Visualization Toolkit*", http://www.bioimagesuite.org, (2006)

[8] Erco's FLTK Cheat Page, http://www.seriss.com/people/erco/fltk/, (2008)

[9] Kitware, Inc., "*Cross-Platform, Open-Source Build System*", http://www.cmake.org/, (2008)

[10] Klinik für Epileptologie, Universität Bonn , „*Röntgen Computertomographie (CT) Hounsfield, 1969 Lehrnetz CT1*", http://www.meb.uni-bonn.de/epileptologie/cms/upload/homepage/lehnertz/CT1.pdf, (2008)

[11] William E. Lorensen, Harvey E. Cline, "*Marching Cubes: A high resolution 2D surace construction algorithm General Electric Company Corporate Research and Development Schenectady"*, Computer Graphics, Volume 21, Number 4, New York 12301 (1987)

[12] DICOM (Digital Imaging and Communications in Medicine), http://dicom.nema.org/, http://de.wikipedia.org/wiki/DICOM, (2008)

[13] Prof. Dr. Olaf Dössel, „*Bildgebende Verfahren in der Medizin. Von der Technik zur medizinischen Anwendung*" , Universität Karlsruhe Institut für Biomedizinische Technik, Springer Verlag, ISBN: 3540660143, Seite: 140-141, (2000)

[14] Uwe Jörg Rother, „*Moderne bildgebende Diagnostik in der Zahn-, Mund- und Kieferheilkunde: Grundlagen, Strahlenschutz, Befunde*", Urban & Fischer in Elsevier Verlag, 2. Auflage, ISBN: 9783437055508, Seite: 166, (2006)

[15] Tcl Developer Xchange, http://www.tcl.tk/, (2008)

[16] Sun Microsystems, http://java.sun.com/, (2008)

[17] Python Software Foundation, http://www.python.org/, (2008)

[18] Kitware, Inc., http://www.kitware.com/, (2008)

[19] Fedora Projekt, http://fedoraproject.org/, http://docs.fedoraproject.org/release-notes/f7/en_US/, (2007)

[20] Dan Pilone & Neil Pitman, „*UML 2.0 in a Nutshell*", O'Reilly Verlag GmbH & Co.KG, 1. Auflage, ISBN: 9783897213425, Seite: 12-35, (2006)

[21] August-Wilhelm Scheer, „*ARIS Modellierungsmethoden, Metamodelle, Anwendungen*", Springer Verlag, 4. Auflage, ISBN: 3540416013, Seite: 106, (2001)

[22] GCC, the GNU Compiler Collection, http://gcc.gnu.org/, (2008)